

UNIVERSITY OF STIRLING

DOCTORAL THESIS

A Novel Sound Reconstruction
Technique based on a Spike Code
(event) Representation

Author:

Madhurananda PAHAR

Supervisor:

Prof. Leslie SMITH

*A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy*

in the

Division of Computer Science and Mathematics

March 2016

Declaration of Authorship

I, Madhurananda PAHAR, declare that this thesis titled, ‘A Novel Sound Reconstruction Technique based on a Spike Code (event) Representation’ and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

UNIVERSITY OF STIRLING

Abstract

Division of Computer Science and Mathematics

Doctor of Philosophy

A Novel Sound Reconstruction Technique based on a Spike Code (event) Representation

by Madhurananda PAHAR

This thesis focuses on the re-generation of sound from a spike based coding system. Three different types of spike based coding system have been analyzed. Two of them are biologically inspired spike based coding systems i.e. the spikes are generated in a similar way to how our auditory nerves generate spikes. They have been called AN (Auditory Nerve) spikes and AN_Onset (Amplitude Modulated Onset) spikes. Sounds have been re-generated from spikes generated by both of those spike coding technique. A related event based coding technique has been developed by Koickal and the sounds have been re-generated from spikes generated by Koickal's spike coding technique and the results are compared.

Our brain does not reconstruct sound from the spikes received from auditory nerves, it interprets it. But by reconstructing sounds from these spike coding techniques, we will be able to identify which spike based technique is better and more efficient for coding different types of sounds.

Many issues and challenges arise in reconstructing sound from spikes and they are discussed. The AN spike technique generates the most spikes of the techniques tested, followed by Koickal's technique (54.4% lower) and the AN_Onset technique (85.6% lower). Both subjective and objective types of testing have been carried out to assess the quality of reconstructed sounds from these three spike coding techniques. Four types of sounds have been used in the subjective test: string, percussion, male voice and female voice. In the objective test, these four types and many other types of sounds have been included. From the results, it has been established that AN spikes generates the best quality of decoded sounds but it produces many more spikes than the others. AN_Onset spikes generates better quality of decoded sounds than Koickal's technique for most of sounds except choir type of sounds and noises, however AN_Onset spikes produces 68.5% fewer spikes than Koickal's spikes. This provides evidences that AN_Onset spikes can outperform Koickal's spikes for most of the sound types.

Acknowledgements

I truly acknowledge all the helps I received from my principle supervisor Prof. Leslie Smith. The Spike based coding technique was designed by him and he inspired me by his ideas on reconstruction technique from those spikes. His helps were invaluable towards the completion of my PhD thesis.

Dr Michael J Newton, present member of acoustic and audio group of University of Edinburgh helped me to run the spike codes in MATLAB and other MATLAB tools while he was in University of Stirling when I started my research. Thomas Jacob Koickal provided his MATLAB code to generate the spikes according to his spike coding technique as mentioned in chapter 2.

I was a self-funded student but I really acknowledge the financial helps from charitable trusts: - ‘The McGlashan Trust’, ‘The Sidney Perry Foundation’ & ‘The CAM Sym Charitable Trust’ each financial year. The division and the university put some teaching job in my way which helped me financially as well.

The teachers in division of computer science and mathematics were exceptionally helpful in my research. Catherine A Howie, teaching fellow & statistical consultant of University of Stirling helped me to develop the statistical model mentioned in chapter 5. I also thank all the volunteers who participated in my subjective testing mentioned in chapter 5.

Also, I lived with my close friends and relatives who looked after me very well which in turn helped me to concentrate into my research.

My parents always motivated me with their kind words and inspired me especially when I went through difficult times. I also thank my God ‘Jehovah’ who gifted me enough intelligence and provided assistance in right times so that I can commence and complete my research successfully.

Dedicated to my mom & dad and true friends . . .

Contents

Declaration of Authorship	i
Abstract	ii
Acknowledgements	iii
Contents	v
List of Figures	x
List of Tables	xxii
Abbreviations	xxvi
1 Introduction	1
1.1 Thesis Inspiration: Coding Sound with spikes	1
1.1.1 What is a spike and an event?	1
1.1.2 Early days of sound coding	1
1.1.3 Lossy-coding technique	3
1.1.4 Why Spike coding has been chosen?	4
1.1.5 Advantages of spike coding over other sound coding techniques . .	5
1.2 Thesis: Aims and Objective	6
1.3 Research Questions	7
1.4 Thesis Outline	8
2 Literature Review	10
2.1 Background	10
2.1.1 Basic Structure and Functions of The Auditory System	10
2.1.1.1 The Outer and Middle Ear	11
2.1.1.2 The Inner Ear and the Basilar Membrane	12
2.1.1.3 The Transduction Process and the Hair Cells	15
2.1.2 Neural Responses and Firing Rates in the Auditory Nerve	16
2.2 The Gammatone Filterbank	18
2.3 Spikes	22
2.3.1 What is a Spike?	22

2.3.2	AN Spike Code	23
2.3.3	Onset Spike Code	28
2.3.4	Koickal's Event Based Spike Code	31
2.4	Other Existing Sound Coding Techniques	37
2.4.1	MP3	37
2.4.1.1	MP3 coding technique	37
2.4.1.2	Comparison with Spike Coding Technique	37
2.4.2	MP4 & AAC	40
2.4.2.1	MP4 & AAC coding technique	40
2.4.2.2	Comparison with Spike Coding Technique	42
2.4.3	WMA (Windows Media Audio)	42
2.4.3.1	WMA Coding Technique	42
2.4.3.2	Comparison with Spike Coding Technique	43
2.5	Background of Subjective Sound Testing	43
2.6	Background of Objective Sound Testing	43
2.6.1	Perceptual Evaluation of Speech Quality Test	43
3	Reconstructing sound from AN Spikes	48
3.1	The Purpose of De-coding from Spike Code	48
3.2	Resynthesis Algorithm	48
3.3	The effect of Delay Vectors	52
3.3.1	Investigating the nature of Delay Vectors	53
3.3.2	No Delay Compensation	58
3.4	Filterbank Tuning for Low sampled Sound	59
3.5	Number of Channels and Sensitivity Labels	65
3.6	Maximum Spiking Rates (MSR)	66
3.6.1	Jittering the AN Spikes	72
3.7	Further Issues in De-coding of AN Spikes	75
3.7.1	Ramp Technique	75
3.7.2	Smoothing Technique	79
3.7.3	Spike Gaps	81
3.7.4	A Single Spike in a Channel	82
3.8	Processing Time	83
3.9	Channel-to-Channel Comparison of Decoding Technique	85
3.9.1	Detailed Comparison of various Reconstructed Sound	85
3.10	Brief Summary of this Chapter	88
4	Reconstructing Sound from Onset Spikes	89
4.1	Amplitude Modulation	89
4.2	Introduction of AN_Onset Spikes	90
4.3	Modification of Onset Spike Generating Parameters	92
4.4	Three Major Types of Combination of AN and Onsets Spikes	95
4.4.1	A brief introduction	95
4.4.2	Reconstructing signal from <i>AN spikes</i>	96
4.4.3	Reconstructing signal from <i>AN_Onset spikes</i>	96
4.4.4	Reconstructing signal from <i>Original Onset spikes</i>	100

4.4.5	Summing up signals from <i>AN spikes</i> , <i>AN-Onset spikes</i> & <i>Original Onset spikes</i>	103
4.5	Tuning the Decoding technique for Three Different types of Sound	104
4.6	Issues in Sound Reconstruction from Onset Spikes	105
4.6.1	Normalization of decoded signals for different frequency levels	105
4.6.2	Large time-gap between two adjacent spikes	105
4.6.3	Effect of Delay vectors	106
4.6.4	Effect of Ramp Technique	107
4.7	Discussion on this Decoding Technique	108
4.8	De-coding from ‘Koickal’s Technique’	109
4.9	Brief Summary of this Chapter	109
5	Subjective Testing & Findings	111
5.1	The Purpose of Testing	111
5.2	Test Equipments Used	112
5.2.1	HTML & Audio Played in Web and Computer	112
5.2.2	Soundproof Room and Headphones	112
5.2.3	Advertisements	112
5.2.4	Volunteers	113
5.2.5	Forms	113
5.2.6	Software	113
5.3	Test Procedure	113
5.4	Background of Testing	114
5.4.1	Three Major Different Coding Techniques used in the Test	114
5.4.2	Sounds used in Testing	115
5.4.2.1	Celesta Sound (Frequency Level 4)	116
5.4.2.2	Celesta Sound (Frequency Level 7)	117
5.4.2.3	Electric Guitar Sound	118
5.4.2.4	Temple Bell	119
5.4.2.5	Male Voice	120
5.4.2.6	Female voice	121
5.5	The Testing Questions and the techniques compared in each questions	121
5.5.1	Question 1: Original Sound vs AN-Decoded Sound	122
5.5.2	Question 2: Original Sound vs AN version of Onset-Decoded Sound	123
5.5.3	Question 3: AN version of Onset-Decoded Sound vs Original	124
5.5.4	Question 4: AN-Decoded Sound vs Original	125
5.5.5	Question 5: AN-Decoded Sound vs AN version of Onset-Decoded Sound	126
5.5.6	Question 6: Original vs AN version of Onset-Decoded Sound	127
5.5.7	Question 7: AN-Decoded Sound vs Koickal’s Reconstructed Sound	128
5.5.8	Question 8: Original vs Koickal’s Reconstructed Sound	129
5.5.9	Question 9: AN version of Onset-Decoded Sound vs Koickal’s Reconstructed Sound	130
5.5.10	Question 10: Original vs AN version of Onset-Decoded Sound	131
5.5.11	Question 11: Koickal’s Reconstructed Sound vs AN-Decoded Sound	132

5.5.12	Question 12: AN version of Onset-Decoded Sound vs AN-Decoded Sound	133
5.5.13	Question 13: AN-Decoded Sound vs AN version of Onset-Decoded Sound	134
5.5.14	Question 14: AN version of Onset-Decoded Sound vs AN-Decoded Sound	135
5.5.15	Question 15: AN-Decoded Sound vs Original	136
5.5.16	Question 16: Original Sound vs AN-Decoded Sound	137
5.5.17	Question 17: Koickal's Reconstructed Sound vs Original Sound . .	138
5.5.18	Question 18: Original Sound vs Koickal's Reconstructed Sound . .	139
5.5.19	Question 19: AN-Decoded Sound vs AN version of Onset-Decoded Sound	140
5.5.20	Question 20: Koickal's Reconstructed Sound vs Original Sound . .	141
5.6	The Statistical Methods used to explain the answers from the Sound Test	142
5.7	The Explanation of answers from each questions and Evidences	143
5.7.1	Question 1: Original Sound vs AN-Decoded Sound	144
5.7.2	Question 2: Original Sound vs AN Version of Onset-Decoded Sound	145
5.7.3	Question 3: AN Version of Onset-Decoded Sound vs Original . . .	146
5.7.4	Question 4: AN-Decoded Sound vs Original	147
5.7.5	Question 5: AN-Decoded Sound vs AN version of Onset-Decoded Sound	148
5.7.6	Question 6: Original vs AN version of Onset-Decoded Sound . . .	149
5.7.7	Question 7: AN-Decoded Sound vs Koickal's Reconstructed Sound	150
5.7.8	Question 8: Original vs Koickal's Reconstructed Sound	151
5.7.9	Question 9: AN version of Onset-Decoded Sound vs Koickal's Reconstructed Sound	152
5.7.10	Question 10: Original vs AN version of Onset-Decoded Sound . . .	153
5.7.11	Question 11: Koickal's Reconstructed Sound vs AN-Decoded Sound	154
5.7.12	Question 12: AN version of Onset-Decoded Sound vs AN-Decoded Sound	155
5.7.13	Question 13: AN-Decoded Sound vs AN version of Onset-Decoded Sound	156
5.7.14	Question 14: AN version of Onset-Decoded Sound vs AN-Decoded Sound	157
5.7.15	Question 15: AN-Decoded Sound vs Original	158
5.7.16	Question 16: Original Sound vs AN-Decoded Sound	159
5.7.17	Question 17: Koickal's Reconstructed Sound vs Original Sound . .	160
5.7.18	Question 18: Original Sound vs Koickal's Reconstructed Sound . .	161
5.7.19	Question 19: AN-Decoded Sound vs AN version of Onset-Decoded Sound	162
5.7.20	Question 20: Koickal's Reconstructed Sound vs Original Sound . .	163
5.8	Sound Testing Conclusion and Remarks	164
5.9	Discussion of Sound Testing Conclusion in Light of Number of Spikes . .	166
5.10	Possible Future Sound Testing and Pros and Cons	170

6.1	The Purpose of Objective Testing	171
6.2	Test Equipments & Procedure	171
6.3	PESQ Results for Subjective Tests	172
6.4	Findings from PESQ on: the five sounds in subjective test	177
6.5	Findings from PESQ on: Number of Channels and Sensitivity levels	180
6.6	Findings from PESQ on Onset Parameters	186
6.7	Findings from PESQ on: Resolution Bits in Koickal's Spike Coding	188
6.8	Findings from PESQ on: Musical Notes with Different Frequencies	189
6.8.1	Celesta	190
6.8.2	Harp	191
6.8.3	Guitar	192
6.8.4	Pianos	193
6.8.5	Woodwinds	194
6.8.6	Gemshorn	195
6.9	Findings from PESQ on: a Choir	197
6.10	Findings from PESQ on: Sounds from Nature	199
6.11	Findings from PESQ on: Reverb Challenge	200
6.12	Findings from PESQ on: Noise	202
6.13	Findings from PESQ on: Signal-to-Noise Ratio (SNR)	204
6.14	Findings from variations of PESQ and Composite Scores & Speech Recognition	210
6.15	Findings from PESQ on: Male Voice & Female Voice	218
6.16	Possible Future Sound Testing and Pros and Cons	218
7	Conclusion	219
7.1	Summary of the Research Work	219
7.1.1	Technique 1: AN Spike Coding	220
7.1.2	Technique 2 & 3: Onset Spike Coding & Koickal's Spike Coding	222
7.1.3	Testing	222
7.2	Empirical Findings	222
7.3	Thesis Contribution & Limitations	223
7.4	Future Works & Final Remarks	227
A	MATLAB Codes for Reconstruction of Sounds from AN and AN_Onset spikes & Other Issues	230
B	Reducing Spike Generation Processing Time (MATLAB Code)	232
C	Subjective Testing HTML Codes & Forms	239
D	Perceptual Evaluation of Speech Quality & Composite Test (MATLAB Codes)	248
	Bibliography	265

List of Figures

2.1	The Whole structure of Auditory System of Mammals (Humans). The position of Ear and part of the head involved with the sound has been described. The External, Middle and Inner Ear and with all its components have been shown in this detailed picture. Source: [24]	11
2.2	A cut-away view of the early auditory system showing the parts which is very similar like human. Source: [26]	12
2.3	The displacement pattern moves from left to right denoting rapid decay after the point of maximal displacement. The dotted line represents the envelop made up by cochlea models invented by von Bekesy. Source: [27]	13
2.4	The instantaneous displacement of BM with six different input frequency. The maximum amplitude has been recorded as 1 and the displacement from the stapes has been measured in millimeter. The displacements have been mentioned with respect to APEX and BASE. Source: [33]	15
2.5	The cross section of Cochlea showing the BM, the tectorial membrane and Organ of Corti. Source: [34]	16
2.6	This is the cross-section of Cochlea where different parts are discussed in the text. Source: [35]	17
2.7	Type I and Type II nerve fibers: The functions of these two neurons has been shown here. Type I fibers (make up about 90-95% of neurons) are connected with inner hair cells and Type II fibers (make up about 5-10% of neurons) are connected with the outer hair cells. Source: [36]	18
2.8	The uncoiled BM, showing the length and the most receptive frequencies in different places on it. Source: [41]	19
2.9	The uncoiled BM similar to figure 2.8. Showing the frequency plot of the most receptive input-sounds. Source: [42]	20
2.10	The uncoiled BM with its frequencies producing maximum vibration. Source: [43]	20
2.11	The spike generation at a glance: In this biologically inspired spike based system, the spikes are generated for each positive-going zero crossing of the input signal. It should also be noticed that the spikes are generated for different sensitivity levels depending on the amplitude of the signal. So, for the high amplitude, which is LOUD, the spikes occur at all the sensitivity levels, but for the low amplitude, which is QUIET, the spikes occur for low sensitivity levels.	24

2.12	AN Bandpass Filter: The microphone signal is taken at a sampling rate of F_S and passed on to this bandpass filter. There are j numbers of filters and the output from each bandpass filters S^j for band j is passed on to the spike coder. For each positive zero crossing, a spike is generated as S_i^j , where i is the number of sensitivity level. This figure has been shown in ([4]).	25
2.13	The construction of AN spikes from a sound: The sound file is $S(t)$. It is passed through the GMF and it splits the signal up into N channels & $S(t)$ becomes to $S_i(t)$, where $1 \leq i \leq N$. The sensitivity levels are introduced to record the amplitude of the signal at the occurrence of each spike. Total number of sensitivity levels used in this coding technique is ξ , where $1 \leq j \leq \xi$. Then the spike generation system generates spike trains for each channel and sensitivity level, which are sequence of times. So, $P(i, j) = \{t_n^{ij}\}$.	26
2.14	The onset generating parameters: Here are three parameters α , β and g , which are responsible for generating the onset spikes. These parameter's values have been investigated in chapter 4.	29
2.15	The onset generation (source [3]): The spikes are generated here only for three band and the depressing synapses and onset generation has been shown only for a single level for those three bands.	29
2.16	The onset spikes are generated from a 6 kHz signal (source: [3]): The rise time is 0.5 millisecond or 3 cycles. The top figure shows the sound and the bottom one shows the occurrence of onsets as the sound is attenuated in dB, varying in 6 dB steps.	30
2.17	Koickal's spike coder: Here spike generation has been explained. The time starts from t_0 till t_1 . $z(t)$ is the feedback signal which is forced to track input signal $x(t)$. The bounding error is $e(t)$. $y(t)$ is the coder output. δ is the tracking step and Δt_d is the time interval between two successive spikes. T is the spike 'width'. N_p is the positive going spikes and N_n is the negative going spikes here. Source: [5]	33
2.18	Koickal's Spike Coder Output for δ (difference between adjacent values) = 3 : In the top figure, the input signal is a speech (in green) and the decoded signal has been marked in red. Here, the decoded signal follows the original signal quite well. From the bottom figure, the generation of spike codes can be explained. When the decoded signal (in red) having a upward movement, there is a positive spike and for a negative movement, there is a negative spike. The sound is a 16 bit so the range of signal is from $-\frac{2^{16}}{2}$ to $(\frac{2^{16}}{2} - 1)$ i.e. from -32768 to 32767.	34
2.19	Koickal's Spike Coder Output for $\delta = 3$: Here the error function can also be seen in the middle figure. The very top and the bottom figure duplicates the figure 2.18.	35
2.20	Koickal's spike coder output for $\delta = 3$ (without low pass filter) : Here few high frequency contents are present. (created in Audacity)	36
2.21	Koickal's spike coder output for $\delta = 3$ (low pass filtered at 3000 Hz) : There are no frequency content higher than 3000 Hz. This sound signal sounds similar with the previous one without low-passed filtered. (created in Audacity)	36
2.22	MP3 Encoding: This encoding process is very complex. The simplified version of encoding is shown here. Source: [20, Figure 6.1]	38

2.23	MP3 Decoding: This process is very complex as well like encoding process. This is the simplified version of decoding. Source: [20, Figure 7.1]	39
2.24	An equal-loudness contour: The peak sensitivity is around 2 - 4 kHz. This is the frequency where the human voice centers the most. Source: [56]	40
2.25	MP4 Encoding process: Source: [60, Figure 1]	41
3.1	The re-construction of sound from AN spike code:- The spike trains for each channel and sensitivity level are received from the AN spike code, $P(i, j) = \{t_n^{ij}\}$, where i represents channels and j represents sensitivity levels. Then they are assigned according to each channel by combining their sensitivity levels and using the highest sensitivity level. So, $P(i, j) = \{t_n^{ij}\}$, becomes $Q(i) = \{t_n^i\}$, where $t_n^m = (\hat{t}, \hat{j})$ for \hat{t} = time and \hat{j} = occurrence in highest sensitivity level of that spike with occurrences for sensitivity level $1 \leq \hat{j} \leq \xi$ (ξ is the total number of sensitivity levels). The sine waves are created for each channel according to each occurrence of spikes. At the end, the regenerated signals for each channels are summed to get back the reconstructed sound signal $S^R(t)$.	49
3.2	The plots of delay vectors by their values along with different filterbank channels, which increases with the increase of frequency. The channel frequency range is from 100 Hz to 10 kHz.	53
3.3	The delay vector plotting for reconstructed signal (without delay compensation):- Here it can be seen that the characteristic of the delays occur at the highest peak of amplitude of the reconstructed signal, roughly speaking. The step function steps from 0 to 1 at time 1.1338 sec. The red crosses(x) [delay point] represent the time value of the delay vector for each channel (see equation 3.11). The delay point is calculated by adding the time value of the zero crossing of the step signal and the delay vector in that particular channel. The green line at the bottom is the original step signal with the value as 1. It can be clearly seen that the 'red cross' [delay point] appears close to the highest peak of the amplitude of the each channel signal.	54
3.4	Comparison between reconstructed signal and the original signal (at the beginning): The reconstructed non-zero signal starts at 0.217 sec, whereas the non-zero original sine wave starts at 0.2267 sec. There is a 0.0097 sec error at the beginning, due to the nature of the delay vector and the reason discussed above. Significantly, it can be seen that before 0.2267 sec, the reconstructed signal has quite a smooth signal curve. This is because that low frequency signal has bigger delay than high frequencies. So, in the delay compensation work, the lower frequency ones come forward than the higher ones.	55
3.5	Comparison between reconstructed signal and the original signal (at the beginning): This time a natural sound like sound of cracking woods has been considered for comparison. The reconstructed non-zero signal starts at 0.2507 sec, whereas the non-zero original sound signal starts at 0.261 sec. There is a 0.0103 sec error at the beginning, due to the nature of the delay vector and the reason discussed above. Significantly, it can be seen that before 0.261 sec, the reconstructed signal has quite a smooth signal curve. This is because that low frequency signal has bigger delay than high frequencies. So, in the delay compensation work, the lower frequency ones come forward than the higher ones.	56

3.6	Comparison between reconstructed signal and the original signal (at the end): The non-similarities are present at the end as well. But this is not because of the Delay Compensation work. it is because the filter ‘rings’ a little.	57
3.7	Comparison between reconstructed signal and the original signal at the beginning (without delay compensation):- Here, we can notice that the decoded sound has started along with the 1 kHz original sound, unlike figure 3.4	58
3.8	Spectrogram of the original sound: This spectrogram, taken from Audacity software for a female speech sound, shows that no the energy contents higher than 8 kHz, are present in the spectrogram. As the sounds are sampled at 16 kHz, the maximum frequency content in those sounds should be 8 kHz. The X-axis shows frequency ranges from 500 Hz to 8000 Hz. Y-axis shows sound intensity from -64 dB to -25 dB.	60
3.9	Spectrogram of the decoded sound of figure 3.8 according to Nyquist frequency in equation 3.12: This spectrogram from Audacity software shows that the energy contents are overly-distributed at the higher frequencies. The X-axis shows frequency ranges from 500 Hz to 8000 Hz. Y-axis shows sound intensity from -69 dB to -25 dB. That’s why, the decoded signal has extra high frequency noises present in them. This shows that in this case GMF is not properly tuned for low sample rates.	61
3.10	Spectrogram of the decoded sound of figure 3.8 according to Adjusted frequency in equation 3.13:- This spectrogram from Audacity software shows that the energy contents are below 8 kHz and the contents are better distributed at the higher frequencies. The X-axis shows frequency ranges from 500 Hz to 8000 Hz. Y-axis shows sound intensity from -78 dB to -24 dB. This sounds much more similar to the original sound.	62
3.11	Spectrogram of the bandpassed signal at the highest frequency according to the adjusted frequency in equation 3.13: This spectrogram from Audacity software shows that the energy contents are at its peak at 4kHz. However, the energy continues up to 5 kHz. The X-axis shows frequency ranges from 500 Hz to 8000 Hz. Y-axis shows sound intensity from -90 dB to -48 dB. This causes to generate the decoded sound without any noise.	63
3.12	Spectrogram of the bandpassed signal at the highest channel (frequency) according to Nyquist frequency in equation 3.12: This spectrogram from Audacity software shows that the energy contents are at its peak at 8 kHz. However, the original sound does not have any energy contents higher than 8 kHz. This causes to generate extra energy contents in the decoded signal shown in figure 3.9, which in fact generates extra noises in the decoded signal as mentioned in figure 3.9.	64

- 3.13 Maximum Spiking Rates:** This technique sorts out the frequency of occurrences of spike rate in a particular channel. t_k are the unsorted spikes and $t_{\hat{k}}$ are sorted spikes, where $k = 1, 2, \dots, 5$ and $\hat{k} = 1, 2, 3$. In this figure it can be seen that the spikes are recorded only after a certain time gap. That time gap has been calculated as 200 Hz, the maximum possible spike firing rate of human auditory nerve. It should be noticed that the sine waves are generated in the center frequency f_c^i in i th channel. But they are multiplied by the ramping multiplier \bar{J}_{k-1}^i to \bar{J}_k^i (see equation 3.4). This technique is very lossy. 67
- 3.14 The original sound signal:** The original sound. Listen to it here: ‘Sound Files Under Test/Speech/Test File (My Name)/testfile.wav’. . . . 69
- 3.15 The decoded sound signal:** The MSR has been applied to the original sound. Listen to it here: ‘Sound Files Under Test/Speech/Test File (My Name)/testfile_16_50_NEW_JTR0.wav’. Here we can see that the energies have been concentrated at certain frequencies. Figure 3.13 explains why this is the case. 69
- 3.16 The insight of the MSR:** $\tilde{f}_{c_k}^i$ is the frequencies for each sine wave generated in normal reconstruction. Here $\tilde{f}_{c_k}^i \sim f_c^i$ (f_c^i is the center frequency of GMF). But, $\bar{f}_{c_k}^i$ is the calculated frequencies, which are almost similar to f_c^i or $\bar{f}_{c_k}^i \simeq f_c^i$ ($k = 2, 3, \dots, 5$ & $\hat{k} = 2, 3$). $\bar{f}_{c_k}^i$ is much closer to f_c^i than $\tilde{f}_{c_k}^i$. This fact explains why there are some extra energy bands in certain frequencies. So, in the reconstructed sound there are some extra energy densities at the frequency level of each channel. 70
- 3.17 The original sound signal:** The original sound is a Human Speech found at ‘Sound Files Under Test/Speech/Test File (My Name)/testfile.wav’. The top figure is the spectrogram and the bottom one is the signal amplitude of that sound. 71
- 3.18 The reconstructed sound signal (With MSR technique):** The reconstructed sound of that original sound found at ‘Sound Files Under Test/Speech/Test File (My Name)/testfile_16_50_NEW_JTR0.wav’. The sound is reconstructed with applying the MSR and Ramp Technique. 71
- 3.19 Not Jittered at all :** This figure is similar with figure 3.15. The MSR has been applied to the original sound. Listen to it here: ‘Sound Files Under Test/Speech/Test File (My Name)/testfile_16_50_NEW_JTR0.wav’ 73
- 3.20 Jittered by 1%:** The frequencies are randomly jittered by 1%. So, the spectrogram gets rid of some extra energy contents in compare to the figure 3.19. But does it sound better than before? Listen to it here: ‘Sound Files Under Test/Speech/Test File (My Name)/testfile_16_50_NEW_JTR2(1%).wav’ 73
- 3.21 Jittered by 10%:** The frequencies are jittered by 10% and the concentrated energy contents almost spreads everywhere in this spectrogram. But it does not help the sound to be better or similar to the original sound signal at all. If we take a closer look to this spectrogram and compare to the original sound’s spectrogram, we will see that there is a pattern in the energy contents in the original sound. But jittering just randomly scatters the extra energy contents everywhere. We do want to get rid of the extra energy contents, but certainly not decreasing the sound quality. Listen to it here: ‘Sound Files Under Test/Speech/Test File (My Name)/testfile_16_50_NEW_JTR2(10%).wav’ 74

3.22 Ramp Technique: In the first case, the sensitivity levels of two consecutive spikes are quite different. So, there are some sharp changes at the zero crossings leading to extra harmonics. The second one, the ramp technique is applied. The discontinuities disappear, as instead of multiplying the whole sine wave by a number, a gradually ramping sequence (\bar{J}_k^i in equation 3.20) is multiplied by the sine wave. The length of these arrows representing spikes increases with the number of sensitivity levels in which the spike occurred.	75
3.23 Spectrogram of the reconstructed sound signal (without MSR technique, with ramp technique) which is similar with the original sound in figure 3.14.	77
3.24 Spectrogram of the reconstructed sound signal (without MSR technique, without ramp technique). This time the spectrogram confirms that reconstructed sound is much similar with the original sound.	78
3.25 Smoothing Technique: Extra sine waves are added at the beginning and at the end to smooth the signal, which reduces extra harmonics energy at the beginning and end of the sound.	79
3.26 Spike Gaps: For a large time gap between the occurrences of two consecutive spikes, the silence in the reconstructed signal has been made. Two extra sine waves have been added in between the silence to reduce the possibility of unwanted harmonics occurring.	81
3.27 A Single Spike:- If there is only one spike in a particular channel, trivially two sine waves are created and concatenated together at the occurrence of the spike. The ramp technique is also applied in this case, to reduce the sharpness in the reconstructed signal.	82
3.28 Original Technique: Construction time of spike code of a sound.	83
3.29 Original Technique: Re-construction time of that sound from its spike code.	84
3.30 New Technique: Construction time of spike code of a sound.	84
3.31 New Technique: Re-construction time of that sound from its spike code.	84
3.32 Spectrogram of the sound signal (directly from the GMF) for the lowest channel frequency.	86
3.33 Spectrogram of the final reconstructed sound signal (reconstructed from spike coding) for that lowest frequency channel mentioned in figure 3.32. In comparison with the previous figure, there are lots of extra energy contents for that single channel signal. This happens because the spike coding technique is a lossy coding technique and that's why the reconstructed signal cannot sound exactly as the original sound. The reconstructed sound adds harmonics, even if they have been minimized.	86
3.34 Spectrogram of the sound signal (directly from the GMF) for a high frequency channel. We can see that the energy contents tend to concentrate to the center frequency of GMF for that channel.	87
3.35 Spectrogram of the reconstructed sound signal (reconstructed from spike coding) for that high frequency channel mentioned in figure 3.34. In comparison with the previous figure, the energy contents scatter far away from the GMF center frequency for that channel. That's why the reconstructed sound does not exactly sounds like original sound.	87

4.1	The top signal is the carrier signal (f_c^i) and the middle one is the modulated signal (f_m^i). The bottom one is the amplitude modulated signal modulated at f_m^i and carried by f_c^i , where $f_c^i > f_m^i$. Source: [78]	90
4.2	Amplitude and Spectrogram plot of the amplitude modulated signal, modulated at 120 Hz with a carrier frequency of 1500 Hz.	91
4.3	Here the x-axis represents time and each set of axes has the y-axis being the channel number (low to high frequency). Those sets are ordered from most sensitive to least sensitive. The first second of a 2 seconds long male speech has been used with the parameter values α , β and g : 100, 9 and 1100 respectively. The figure shows the appearances of original onset spikes (black dots) for different sensitivity levels for a speech signal. It can be seen that the onset spikes appear more in lower sensitivity levels but fewer in higher sensitivity levels.	92
4.4	The figure shows that the onset spikes (blue dots) appear at appropriate times. Also, they are more in numbers, so the parameter values used to generate these onsets can be used for AM detector. Here x-axis is the time line and the y-axis has been the sensitivity levels in ascending order.	93
4.5	The figure shows that the onset spikes (in blue dots) have appeared at each 'tick-tock' sound of the clock. The appearance of these onset spikes is very accurate as well. Here x-axis is the time line and the y-axis has been the sensitivity levels in ascending order.	94
4.6	The re-construction of sound from AN_Onset spikes: The spike trains for each channel and sensitivity level are received from the AN spike code, where $P(i, j) = \{t_n^{ij}\}$, where i represents channels and j represents sensitivity levels. Then they are assigned according to each channel by combining their sensitivity levels and considering the highest sensitivity level. So, $P(i, j) = \{t_n^{ij}\}$, becomes $Q(i) = \{t_n^i\}$, where $t_n^m = (\hat{t}, \hat{j})$ for \hat{t} = time and \hat{j} = occurrence in highest sensitivity level of that spike with occurrences for sensitivity level $1 \leq \hat{j} \leq \xi$ (ξ is the total number of Sensitivity Levels). Then the amplitude modulated signals, modulated at 125 Hz and carried by the filterbank center frequency, are created for each channel according to each occurrence of spikes. At the end, the regenerated signals for each channels are summed up to get back the reconstructed sound signal $S^R(t)$	97
4.7	The re-construction of sound from original onset spikes: The spike trains for each channel and sensitivity level are received from the AN spike code, where $P(i, j) = \{t_n^{ij}\}$, where i represents channels and j represents sensitivity levels. Then they are assigned according to each channel by combining their sensitivity levels and considering the highest sensitivity level. So, $P(i, j) = \{t_n^{ij}\}$, becomes $Q(i) = \{t_n^i\}$, where $t_n^m = (\hat{t}, \hat{j})$ for \hat{t} = time and \hat{j} = occurrence in highest sensitivity level of that spike with occurrences for sensitivity level $1 \leq \hat{j} \leq \xi$ (ξ is the total number of Sensitivity Levels). Then the static white noises are created for each channel according to each occurrence of spikes. At the end, the regenerated signals for each channels are summed up to get back the reconstructed sound signal $S^R(t)$	101

4.8	Decoded signal generated for fewer spike: This is the signal plot of a high frequency channel, where very few original onset spikes appear. The signal plot with yellow color at the background represents the original sound. The green straight lines appeared along the x-axis (which represents the sound length in second) are the occurrences of original onset spikes. Y-axis represents the amplitude values of the sound signal. The blue signals are the actual decoded signal generated for this high frequency channel. These blue sections show that the decoded signals are generated only where the spikes appear.	106
4.9	Ramp Technique for onset spikes: An Amplitude Modulated signal has been generated and then has been multiplied by a ramp multiplier as shown by the blue signals. In the background, the yellow signal is the original sound. X-axis represents the amplitudes and the Y-axis represents time in seconds.	107
5.1	Celesta sound (frequency level 4): This spectrogram shows that the frequency contents are concentrated near 370 Hz. This sound can be heard by clicking here (www.cs.stir.ac.uk/~mpa/testing/questions_1.html) and then by playing the FIRST sound.	116
5.2	Celesta sound (frequency level 7): This spectrogram shows that the frequency contents are concentrated near 2986 Hz and apart from that there are very few frequency contents which are present in the spectrogram. This sound can be heard by clicking here (www.cs.stir.ac.uk/~mpa/testing/questions_2.html) and then by playing the FIRST sound.	117
5.3	Electric Guitar Sound: This spectrogram shows that the frequency contents are distributed very densely from 500 Hz to about 3800 Hz. Apart from that there are few frequency contents which are present in the spectrogram. This sound can be heard by clicking here (www.cs.stir.ac.uk/~mpa/testing/questions_16.html) and then by playing the FIRST sound.	118
5.4	Temple Bell: This spectrogram shows that the frequency contents are distributed very densely from 500 Hz to about 1000 Hz. More frequency contents can be found at 3300 Hz to 4000 Hz and at 4500 Hz to 5000 Hz. This sound can be heard by clicking here (www.cs.stir.ac.uk/~mpa/testing/questions_3.html) and then by playing the SECOND sound.	119
5.5	Male voice: This spectrogram of a male speech shows the typical frequency content distribution in a male speech signal. This sound can be heard by clicking here (www.cs.stir.ac.uk/~mpa/testing/questions_10.html) and then by playing the FIRST sound.	120
5.6	Female Voice: This spectrogram of a female speech shows the typical frequency content distribution in a female speech signal. This sound can be heard by clicking here (www.cs.stir.ac.uk/~mpa/testing/questions_6.html) and then by playing the FIRST sound.	121
5.7	Number of spikes for various sound lengths in seconds: The number of spikes have been divided by 10000 to scale it along with lengths in seconds.	168

5.8	Number of spikes for different sound types: The number of spikes have been divided by 10000 to scale it along with lengths in seconds. The sound length has been measured in seconds.	169
6.1	The test scores of question 1 to 10: The test scores have been plotted together and compared along with their maximum values. The C represents C_{sig} , B represents C_{bak} & O represents C_{owl} ; as explained in chapter 2. The test scores 0 suggests that there is no similarity between the clean and enhanced sound at all. Originally MATLAB produces negative scores for some of them, but they have been normalized as 0 ([85]).	174
6.2	The test scores of question 11 to 20: The test scores have been plotted together and compared along with their maximum values. The test scores 0 suggests that there is no similarity between the clean and enhanced sound at all. Originally MATLAB produces negative scores for some of them, but they have been normalized as 0 ([85]).	176
6.3	PESQ test scores for different types of sounds: Here, the PESQ scores have been compared for the five different sounds and they are shown in this figure according to those five different sound types. ‘PESQ-Thomas’ represents the PESQ scores from Koickal’s code. AN & AN-Onset spike codes produce the highest PESQ test scores for Male Voice and the Celesta Frequency 4 (String1). So, the Male Voice and the Celesta Frequency 4 (String1) has been decoded as the best among others for AN and AN-Onset spike coding.	178
6.4	PESQ Test Scores for different types of codes: Here, the PESQ scores have been compared for the five different sounds and they are shown in this figure according to three different spike coding techniques. Here, ‘Thomas’ represents Koickal’s code. PESQ-a represents ‘Percussion’ sound, ‘b’ represents ‘String 1’, ‘c’ represents ‘String 2’, ‘d’ represents ‘Male Speech’ & ‘e’ represents ‘Female Speech’. The PESQ scores suggest that AN spike code is the best, AN-Onset spike code is better and Koickal’s spike code is the worse for all of these five different types of sounds. . . .	179
6.5	The PESQ test scores of different numbers of channels and sensitivity levels (c_{NSM} means total N channels and M sensitivity levels are used) for BEST PESQ: The composite PESQ test scores have been plotted together and compared along with their maximum values. This figure has been sorted according to the highest PESQ score. The test scores 0 suggests that there is no similarity between the clean and enhanced sound at all ([85]).	180
6.6	The PESQ test scores of different channels and sensitivity levels (c_{NSM} means total N channels and M sensitivity levels are used) for BEST Signal Distortion (C_{sig}): The composite PESQ test scores have been plotted together and compared along with their maximum values. This figure has been sorted according to the highest C_{sig} score. The test scores 0 suggests that there is no similarity between the clean and enhanced sound at all ([85]).	181

6.7	The PESQ test scores of different channels and sensitivity levels ($c_N s_M$ means total N channels and M sensitivity levels are used) for BEST Noise Distortion (C_{bak}): The composite PESQ test scores have been plotted together and compared along with their maximum values. This figure has been sorted according to the highest C_{bak} score. The test scores 0 suggests that there is no similarity between the clean and enhanced sound at all ([85]).	182
6.8	The PESQ test scores of different channels and sensitivity levels ($c_N s_M$ means total N channels and M sensitivity levels are used) for BEST C_{ovl}: The composite PESQ test scores have been plotted together and compared along with their maximum values. This figure has been sorted according to the highest C_{ovl} score and the differences are really quite small. The test scores 0 suggests that there is no similarity between the clean and enhanced sound at all ([85]).	183
6.9	The bytes required to store AN spikes for 16 sensitivity levels & 50 channels: This figure plots the required kilobytes along with the number of spikes for each channel. On the left side, the number of spikes are shown which ranges from 10^2 to 10^4 and on the right side, the required kilobytes are shown which ranges from 0 to 50 kilobytes. For a 2.8706 second sound file, 274.7018 kilobytes of AN spikes are required at 16 sensitivity levels and 50 channels.	184
6.10	The bytes required to store AN spikes for 32 sensitivity levels & 75 channels: This figure plots the required kilobytes along with the number of spikes for each channel. Bytes required for AN spikes of a 2.8706 sec sound file is 409.2827 kilobytes at 32 sensitivity levels and 75 channels.	185
6.11	PESQ test scores for different onset parameters: Here the PESQ and composite scores for 5 different onset-parameter combinations (c1 to c5) are used. The test scores show that combination 1 provides the highest PESQ scores indicating that using that combination of parameter values in coding Onset spikes will provide the best quality of decoded sound.	187
6.12	PESQ test scores for different parameters in Koickal's sound reconstruction: This figure shows that for the resolution parameter value -'7', the PESQ score is the highest. And also, the other composite scores are high for resolution value '7' as well. So, the Resolution parameter value has been set to '7'.	188
6.13	PESQ test scores for different parameters in Koickal's sound reconstruction: This figure shows that for the resolution parameter value -'7', the PESQ score is the highest. So, the resolution parameter value has been set to '7'.	189
6.14	PESQ test scores for Celesta notes: This figure shows the PESQ scores for Celesta notes A4 to A7. 'PESQ-Thomas' represents the PESQ scores from Koickal's code.	190
6.15	PESQ test scores for Harp notes: This figure shows the PESQ scores for Harp notes A3 to A5. 'PESQ-Thomas' represents the PESQ scores from Koickal's code.	191
6.16	PESQ test scores for Guitar notes: This figure shows the PESQ scores for Guitar notes A3 to A6. 'PESQ-Thomas' represents the PESQ scores from Koickal's code.	192

6.17	PESQ test scores for Piano loud notes: This figure shows the PESQ scores for piano loud notes from B0 to B7. ‘PESQ-Thomas’ represents the PESQ scores from Koickal’s code.	193
6.18	PESQ test scores for Woodwinds notes: This figure shows the PESQ scores for woodwinds notes for A2 & A3. ‘PESQ-Thomas’ represents the PESQ scores from Koickal’s code.	194
6.19	PESQ test scores for Gemshorn notes: This figure shows the PESQ scores for Gemshorn notes A3 to A6. ‘PESQ-Thomas’ represents the PESQ scores from Koickal’s code.	195
6.20	PESQ test scores for Choir type of sound: AN[H] represents the ‘hallelujah’ and AN[U] represents the choir song. AN[AVR] represents the average of those two. ‘Thomas[H]’ represents the PESQ scores from Koickal’s code for ‘hallelujah’ etc.	197
6.21	PESQ test scores for Nature type of sound: AN[N] represents the ‘cracking wood noise’ and AN[R] represents the crow song. AN[AVR] represents the average of those two. ‘Thomas[N]’ represents the PESQ scores from Koickal’s code for ‘cracking wood noise’ etc.	199
6.22	PESQ test scores for Reverb Challenges: AN[CS] represents the ‘countdown in a small hall’ and AN[CB] represents that countdown in a big hall. AN[KS] represents the ‘clock tick-tock in a small hall’ and AN[KB] represents that clock tick-tock in a big hall. AN[OVL] represents the average of those two. ‘Thomas[CS]’ represents the PESQ scores from Koickal’s code for ‘countdown in a small hall’ etc.	201
6.23	PESQ test scores for Noises: The PESQ scores are not very consistent with each other. ‘PESQ-Thomas’ represents the PESQ scores from Koickal’s code.	202
6.24	PESQ test scores for SNR 10dB. AN[C] represents the ‘canteen noise’, AN[S] represents the ‘sea-ship noise’ and AN[N] represents ‘static noise’. AN[OVL] represents the average of those three. ‘Thomas[C]’ represents the PESQ scores from Koickal’s code for ‘canteen noise’ etc.	204
6.25	PESQ test scores for SNR 20dB. AN[C] represents the ‘canteen noise’, AN[S] represents the ‘sea-ship noise’ and AN[N] represents ‘static noise’. AN[OVL] represents the average of those three. ‘Thomas[C]’ represents the PESQ scores from Koickal’s code for ‘canteen noise’ etc.	205
6.26	PESQ test scores for SNR 30dB. AN[C] represents the ‘canteen noise’, AN[S] represents the ‘sea-ship noise’ and AN[N] represents ‘static noise’. AN[OVL] represents the average of those three. ‘Thomas[C]’ represents the PESQ scores from Koickal’s code for ‘canteen noise’ etc.	206
6.27	PESQ test scores for SNR (30, 20, 10dB) with background Canteen Noise. AN[30] represents the ‘30 db SNR’, AN[20] represents the ‘20 db SNR’ and AN[10] represents ‘10 db SNR’. AN[OVL] represents the average of those three. ‘Thomas[30]’ represents the PESQ scores from Koickal’s code for ‘30 db SNR’ etc.	207
6.28	PESQ test scores for SNR (30, 20, 10dB) with background Sea-ship Noise. AN[30] represents the ‘30 db SNR’, AN[20] represents the ‘20 db SNR’ and AN[10] represents ‘10 db SNR’. AN[OVL] represents the average of those three. ‘Thomas[30]’ represents the PESQ scores from Koickal’s code for ‘30 db SNR’ etc.	208

6.29	PESQ test scores for SNR (30, 20, 10dB) with background Static Noise. AN[30] represents the ‘30 db SNR’, AN[20] represents the ‘20 db SNR’ and AN[10] represents ‘10 db SNR’. AN[OVL] represents the average of those three. ‘Thomas[30]’ represents the PESQ scores from Koickal’s code for ‘30 db SNR’ etc.	209
6.30	PESQ test scores for male voices: The standard deviation of PESQ scores among all those participants are: 0.247628106 for AN, 0.283736043 for AN-Onset & 0.160185036 for Koickal (PESQ-Thomas)	210
6.31	PESQ test scores for female voices: The standard deviation of PESQ scores among all those participants are: 0.122429292 for AN, 0.068171541 for AN-Onset & 0.092378789 for Koickal (PESQ-Thomas)	211
6.32	C test scores for male voices: The standard deviation of C_{sig} scores among all those participants are: 0.183352546 for AN, 0.477321344 for AN-Onset & 0.349324843 for Koickal (C-Thomas)	212
6.33	C test scores for female voices: The standard deviation of C_{sig} scores among all those participants are: 0.181996051 for AN, 0.312186634 for AN-Onset & 0.210768882 for Koickal (C-Thomas)	213
6.34	B test scores for male voices: The standard deviation of C_{bak} scores among all those participants are: 0.143547001 for AN, 0.160907146 for AN-Onset & 0.15334718 for Koickal (B-Thomas)	214
6.35	B test scores for female voices: The standard deviation of C_{bak} scores among all those participants are: 0.113980518 for AN, 0.082400131 for AN-Onset & 0.116492891 for Koickal (B-Thomas)	215
6.36	O test scores for male voices: The standard deviation of C_{ovl} scores among all those participants are: 0.213113825 for AN, 0.430407418 for AN-Onset & 0.244679199 for Koickal (O-Thomas)	216
6.37	O test scores for female voices: The standard deviation of C_{ovl} scores among all those participants are: 0.149313295 for AN, 0.183288138 for AN-Onset & 0.140653253 for Koickal (O-Thomas)	217
C.1	The Web Output of Question 1:- This is the web output of the code mentioned above. The user has to choose the sound which is better among sound 1a and sound 1b.	242
C.2	The Advert used to gather volunteers for subjective test.	242

List of Tables

2.1	Scale of Signal Distortion (SIG). Source: Table 1, [66]	44
2.2	Scale of Background Intrusiveness (BAK). Source: Table 1, [66]	45
4.1	The GMF frequency distribution for low, middle and high frequency range which have been used in reconstruction of sounds.	104
4.2	The number of spikes increases with the increase in the bit depth for the Koickal coding of a male speech sound of 2 seconds in length.	109
5.1	Test Question 1. Here 11 out of 21 have favored the original sound over the AN-decoded sound. The spectrograms are fairly similar to each other.	122
5.2	Test Question 2. Here 19 out of 21 have favored the original sound over the AN_Onset-decoded sound. This is to be noticed here that the spectrograms of the original and decoded sound are very different from each other. This is because AN_Onset spike coding is not sensible at all for the high frequency sounds. Here a celesta high frequency level 7 has been chosen, so AN_Onset spike coding cannot contain high frequency contents in it.	123
5.3	Test Question 3. Here only 1 out of 21 has favored the AN_Onset-decoded sound over the original sound.	124
5.4	Test Question 4. Here nobody has favored the AN-decoded sound over the original sound.	125
5.5	Test Question 5. Here 19 out of 21 have favored the AN-decoded sound over the AN_Onset-Decoded sound.	126
5.6	Test Question 6. Here all of the participants have favored the original sound over the AN_Onset-Decoded sound.	127
5.7	Test Question 7. Here all of the participants have favored the AN-Decoded sound over the Koickal's Reconstructed sound.	128
5.8	Test Question 8. Here all of the participants have favored the AN-Decoded sound over the Koickal's Reconstructed sound.	129
5.9	Test Question 9. Here 16 out of 21 have favored the AN-Onset-Decoded sound over the Koickal's Reconstructed sound. This is to be noticed here that the spectrograms of the AN_Onset decoded sound and Koickal's decoded sound are very different from each other, similar with question 2. This is because AN_Onset spike coding is not sensible at all for the high frequency sounds. Here a celesta high frequency level 7 has been chosen, so AN_Onset spike coding cannot contain high frequency contents in it.	130
5.10	Test Question 10. Here all of the participants have favored the original sound over the AN-Onset-Decoded sound.	131
5.11	Test Question 11. Here none of the participants have favored the Koickal's Reconstructed sound over the AN-Decoded sound.	132

5.12	Test Question 12. Here 7 of the participants have favored the AN-Onset-Decoded sound over the AN-Decoded sound.	133
5.13	Test Question 13. Here 9 of the participants have favored the AN-Decoded sound over the AN-Onset-Decoded sound.	134
5.14	Test Question 14. Here 3 of the participants have favored the AN-Onset-Decoded sound over the AN-Decoded sound.	135
5.15	Test Question 15. Here nobody has favored the AN-decoded sound over the original sound.	136
5.16	Test Question 16. Here 19 out of 21 have favored the original sound over the AN-decoded sound.	137
5.17	Test Question 17. Here none of the participants have favored the Koickal's Reconstructed sound over the original sound.	138
5.18	Test Question 18. Here all of the participants have favored the original sound over the Koickal's Reconstructed sound.	139
5.19	Test Question 19. Here 14 of the participants have favored the AN-Decoded sound over the AN-Onset-Decoded sound.	140
5.20	Test Question 20. Here none of the participants have favored the Koickal's Reconstructed sound over the original sound.	141
5.21	Test Question 1. The AN-Decoded sound is not different from the Original.	144
5.22	Test Question 2. The AN-Onset-Decoded sound is different from the Original. The statistical evidences also suggest that original sound is better than the AN-Onset-Coded sound.	145
5.23	Test Question 3. The Original sound is different from the AN-Onset-Decoded sound. The statistical evidences also suggest that AN-Onset-Decoded sound is worse than the Original sound.	146
5.24	Test Question 4. The Original sound is different from the AN-Decoded sound. The statistical evidences also suggest that AN-Decoded sound is worse than the Original sound.	147
5.25	Test Question 5. The AN-Decoded sound is different from the AN-Onset-Decoded sound. The statistical evidences also suggest that AN-Decoded sound is better than the AN-Onset-Decoded sound.	148
5.26	Test Question 6. The AN-Onset-Decoded sound is different from the Original. The statistical evidences also suggest that original sound is better than the AN-Onset-Decoded sound.	149
5.27	Test Question 7. The AN-Decoded sound is different from the Koickal-Reconstructed sound. The statistical evidences also suggest that AN-Decoded sound is better than the Koickal-Reconstructed sound.	150
5.28	Test Question 8. The Original sound is different from the Koickal-Reconstructed sound. The statistical evidences also suggest that Original sound is better than the Koickal-Reconstructed sound.	151
5.29	Test Question 9. The AN-Onset-Decoded sound is different from the Koickal-Reconstructed sound in 90 or 95 times out of 100. The statistical evidences also suggest that AN-Onset-Decoded sound is better than the Koickal-Reconstructed sound. However for 99 times out of 100, we do not have such evidences.	152
5.30	Test Question 10. The Original sound is different from the AN-Onset-Decoded sound. The statistical evidences also suggest that Original sound is better than the AN-Onset-Decoded sound.	153

5.31	Test Question 11. The Koickal-Reconstructed sound is different from the AN-Decoded sound. The statistical evidences also suggest that Koickal-Reconstructed sound is worse than the AN-Decoded sound.	154
5.32	Test Question 12. The AN-Onset-Decoded sound is NOT different from the AN-Decoded sound.	155
5.33	Test Question 12. The AN-Decoded sound is NOT different from the AN-Onset-Decoded sound.	156
5.34	Test Question 14. The AN-Onset-Decoded sound is different from the AN-Decoded sound. The statistical evidences also suggest that AN-Onset-Decoded sound is worse than the AN-Decoded sound.	157
5.35	Test Question 15. The AN-Decoded sound is different from the Original sound. The statistical evidences also suggest that AN-Decoded sound is worse than the Original sound.	158
5.36	Test Question 16. The Original sound is different from the AN-Decoded sound. The statistical evidences also suggest that Original sound is better than the AN-Decoded sound.	159
5.37	Test Question 17. The Koickal-Reconstructed sound is different from the Original sound. The statistical evidences also suggest that Koickal-Reconstructed sound is worse than the Original sound.	160
5.38	Test Question 18. The Original sound is different from the Koickal-Reconstructed sound. The statistical evidences also suggest that Original sound is better than the Koickal-Reconstructed sound.	161
5.39	Test Question 19. The AN-Decoded sound is NOT different from the AN-Onset-Decoded sound.	162
5.40	Test Question 20. The Koickal-Reconstructed sound is different from the Original sound. The statistical evidences also suggest that Koickal-Reconstructed sound is worse than the Original sound.	163
5.41	Test Summary: Here the test results have been summed up. This table shows the techniques which are better than or same as the others for each 'string', 'percussion', 'male voice' and 'female voice'.	164
5.42	Number of spikes: Here the number of spikes per second for four different types of sounds have been mentioned along with the corresponding length in seconds. Koickal's spikes are 54.4% lower in number and AN_Onset spikes are 85.6% lower in number than AN spikes. AN_Onset spikes are 68.5% lower in number than Koickal's spikes.	167
6.1	The Objective test scores on the sounds used in the Subjective test for question 1 to question 10. The better sound has been compared with the worse sound i.e. the better sound has been used as clean file and the worse one has been the enhanced file.	173
6.2	The Objective test scores on the sounds used in the Subjective test for question 11 to question 20. The better sound has been compared with the worse sound i.e. the better sound has been used as clean file and the worse one has been the enhanced file.	175

6.3	Number of spikes : Here the number of spikes for two different frequencies of Gemshorn tunes have been mentioned along with the corresponding length in seconds. The numbers for each coding type are the total number of spikes used to decode the reconstructed sound in that coding technique. So, to decode the sound from AN_Onset spikes codes, 111884 spikes have been used for frequency 3 Gemshorn sound.	196
6.4	PESQ test scores for all musical notes: - These are the average PESQ test scores for all the Musical Notes along with their standard deviations.	196
6.5	Number of spikes : Here the number of spikes for two different Choir sounds have been mentioned along with the corresponding length in seconds. The numbers for each coding type are the total number of spikes used to decode the reconstructed sound in that coding technique. So, to decode the sound from AN_Onset spikes codes, 160274 spikes have been used.	198
6.6	These are the average PESQ test scores for all the Noises with their standard deviations.	202
6.7	Number of Spikes : Here the number of spikes for four noises have been mentioned along with the corresponding length in seconds. The numbers for each coding type are the total number of spikes used to decode the reconstructed sound in that coding technique. So, to decode the Oceanwave sound from AN_Onset spikes codes, 151653 spikes have been used.	203

Abbreviations

AN Spikes	Auditory Nerve like Spikes
AN_Onset Spikes	More Frequent Onset Spikes to detect Amplitude Modulation
Koickal's Spikes	Thomas Jacob Koickal's Spikes
PESQ	Perceptual Evaluation of Speech Quality
BM	Basilar Membrane
GMF	Gammatone Filterbank
AM	Amplitude Modulation
AER	Address Event Representation
ASR	Automated speech recognition
MSR	Maximum Spiking Rate
SNR	Signal-to-Noise Ratio
MP3	MPEG-1 or MPEG-2 Audio Layer III, commonly referred as MP3
AAC	Advanced Audio Coding
MP4	a digital multimedia container format most commonly used to store video and audio, with extension .m4a or .m4p
WMA	Windows Media Audio
WAV	Waveform Audio File Format (WAVE or WAV filename extension)
MPEG	Moving Picture Experts Group
AU	an audio file format introduced by Sun Microsystems
AIFF	Audio Interchange File Format
ITD	Intermicrophone Time Difference
IHC	Inner hair cells
PC	Personal Computer
CF	Center Frequency
RMS	Root Mean Square Value

Chapter 1

Introduction

1.1 Thesis Inspiration: Coding Sound with spikes

1.1.1 What is a spike and an event?

The Oxford Dictionary definition of a spike is “a sharp increase in the magnitude or concentration of something” [1]. Because it is localized in time, a spike is useful to describe an event. An event is when something ‘happens or takes place, especially one of importance’ [2]. A spike marks a point in time when an event has occurred. The way they are defined in this thesis may be found in section 2.3 of chapter 2.

In this thesis, the sound reconstruction from its spike coded state and both subjective and objective testing is my work. The spike generation techniques are developed by Leslie Smith and others ([3]) and ([4]). Koickal’s spike coding technique ([5]) has been invented by Dr. Thomas Jacob Koickal.

1.1.2 Early days of sound coding

Listening or hearing is an essential part in our everyday life. Almost 99% of the world’s population listens to radio, TV or some other form of audio playing device ([6]). The business and marketing in this field has changed dramatically in the last few decades. So the research activity in this field is at its peak in recent years. Several audio coding

techniques have been developed and they have been used for both research and marketing purpose.

The idea of coding sound began when the recording for playback began. In the early days in 1900, carbon microphones were used to convert sound to an electrical signal. One of the most popular sound playing devices was Phonograph in 1887. However before phonograph, there was Phonautograph, invented by Leon Scott in 1857. This was the first sound recording device known in the history [7]. However it was able to record sound for only less than 1 second. Then phonographs and graphophones were used for few years until electrical recordings were available in 1920s [8]. Stereo type recordings were available for home-usage in 1930 in form of sound tapes [9]. Then the era of digital recordings and compact disks came in 1980s [10]. Back in 1988, IBM first developed the sound card which was able to produce only certain types of sounds, though their usage was very limited until Sound Blaster cards were brought to market by Creative Technology Limited in November 1990 for multimedia PCs [11].

The idea of coding and decoding compressed digital sound was standardized in the year 1997 when AAC types of sound was first produced. AAC stands for Advanced Audio Codec. This was updated in 1999 by including MPEG-4 part 3 types and some Perceptual Noise Substitution. MP3 is another common audio coding format referring to MPEG-1 and MPEG-2 [12]. AAC is the successor of MP3 and in time AAC has been improved over MP3. The areas where the advancements have been made are important for this research as these will explain some of the advancements made though our biologically inspired spike based coding technique. AAC provides much bigger range of sampling frequencies, accuracies and frame lengths than MP3. These improvements of AAC over MP3 have been described in the next literature review (chapter 2).

So, it is quite clear that different audio coding techniques work differently and of course they have their pros and cons over each other. Three major types of sound coding techniques are uncompressed, lossy and lossless coding [13]. Uncompressed coding is the type of coding where the information in the audio has not been compressed in any form. Lossless compression is where the audio has been compressed but no information has been missed out i.e. all the information can be received by decompression whereas in lossy coding the audio has been compressed and some information has been deliberately lost. In fact, in lossy coding, only the important and relevant information has been stored

in the code. The most common types of uncompressed coding techniques used are WAV, AIFF, AU, WMA lossless (.wma), Monkey's audio (.ape), Apple lossless (.m4a). The commonly used lossy audio coding are: MP3, AAC, WMA lossy. They come into the center of our focus and are illustrated in chapter 2. As the lossless coding technique is where the information in the sound has not been lost by the coding technique, the decoded sound from the lossless coding should be exactly similar to the original sound. This technique is desirable when any loss of information in the original sound cannot be allowed. For example, MPEG-4 SLS is a lossless audio coding technique. Whereas the lossy coding technique is a compression technique where some information in the original sound has been lost. As mentioned before, MP3, AAC, WMA are some examples of lossy coding techniques in the auditory field.

1.1.3 Lossy-coding technique

The question arises here - why do we need lossy coding technique? The primary reason is to reduce file size or to reduce data volume. Also, not all information is required in the decoded sound. For example, if there is a big pause in the sound, not all the bits are required to be coded. The maximum frequency a human can hear is about 20 kHz, which decreases with the increase of age. So, any frequency more than 20 kHz should not be in the coded sound, as we will not be able to hear them. And there are many other areas where the sound coding can be lossy but which will not affect the human auditory system much.

In this research project we are interested only in the lossy coding techniques in particular a biologically inspired spike coding technique to code sound. The technique is event based and lossy. A more detailed comparison between our spike codes and MP3, AAC, WMA is included in chapter 2. Three major types of spike coding techniques are considered in this research and all of them are lossy. Biologically inspired spike codes (AN and AN_Onset spikes) follow closely the human brain activities and the other hardware-based spike code (Koickal's spike code) is event based and they are discussed in the literature review chapter (chapter 2).

1.1.4 Why Spike coding has been chosen?

The spike based coding technique is just another way to code sound. Like many other sound coding techniques, it has its benefits and limitations. Two of our spike coding techniques discussed in this thesis, are biologically inspired, unlike other coding-decoding technique used in electronic chip devices. How our spike coding technique will be helpful to separate different sources of sound, its intelligibility, amplitude modulation or frequency modulation? For example, when we are sitting in a restaurant and talking to each other, we can concentrate on our talk despite the background noise. That suggests that we can robustly separate the sound source or probably the sound stream or sound frequency in complex acoustic senses in some way in our brain. And this spike based coding system should be very helpful to develop a model in computer to do same kind of sound stream segregation.

Finding out the common pattern in onset and offset times, harmonics, amplitude and frequency modulation, pitch or spatial position can help us to recognize the ‘right’ sound stream i.e. sound which we want to listen to. We are confident that this spike based code will help us to provide such patterns in sound streams. By using this biologically inspired spike based coding system, it is possible to determine ‘Intermicrophone Time Difference (ITD)’ from multiple sound sources during Onset Intervals [4] as well. Interestingly direction determination during these Onset Intervals ¹, enables us to find out the azimuthal angle of onsetting sounds even in the presence of competing sounds. In that article [4], three techniques are discussed i.e. ‘Cross-Correlation’, ‘AN-like Spikes’ and ‘Onset Spikes’. Also in that paper, speech with background noise has been tested to measure ITDs and the results say that Onset spike technique almost outperforms the AN-like technique. Again multiple human speeches delayed by 0.3 sec from another has been tested to measure the ITD and this time the results are fairly similar, although onset spike technique provides better valid angle for more onset intervals. One of its authors was talking and moving when a speech sound has been recorded and tested. Both onset and AN-like spike successfully tracked the sound source. The applications of the success of this experiment are huge.

¹Sounds result from vibration which takes time to build up and the sound takes time to reach its maximal intensity. This time is the Onset interval of the sound.

Hearing aid is a potential device to apply the technique. According to Sebastien Santurette, ‘If auditory prostheses can transmit the pitch of sounds to hearing-impaired patients in a better way, this will improve their ability to segregate speech sources and enjoy music’ [14]. A person can sit in the rail station, busy shopping market or in a quiet place and the sounds can be segregated by the hearing aid, so that he gets the sound, not the ‘noise’ [Here, the noise does not necessarily have to be a white noise. Any other sounds apart from the desired sound can be represented by noise.]. Morten Løve Jepsen is confident to say that on the basis of his project, an auditory-model based system could be developed which objectively evaluates the effects of hearing-aid signal processing [15].

In fact by using various mathematical models, it is possible to analyze sound signals that reproduce the auditory features [16]. So, we can understand which kind of representations our higher levels in the brain use to isolate signals from noise, or to separate signals which have different pitches. Lyon’s Auditory Model Inversion [17] uses a mathematical auditory model to resynthesize the same sound. They have analyzed and synthesized speech signal first and then they have synthesized more complex signals (music and vocal signals) and in both case the system has well behaved. Automatic Speech Recognition (ASR) systems [18] can segregate sounds from interfacing speech stream, although some of them do not work well in the presence of competing voices or interfacing noises. So, also in this case, sound segregation is a major issue. Our spike based coding system can be very helpful in ASR like system to make it work more efficiently and effectively. There exists another system called NoiseTracker II, which can use spectral subtraction to get rid of noise in a noisy environment. This is one of the most widely used methods according to figure 9 of [19].

1.1.5 Advantages of spike coding over other sound coding techniques

There are several advantages of using spikes for coding sound. It is true that spikes are neurally inspired. The usage of spike coding as a sound coding technique in synthesis-resynthesis system is that spikes can record precise timing of events and is capable to record the loudness of the signal as well.

There are several audio coding techniques available in these days and they are used in almost every musical digital device. Some of the most popular audio codecs are MP3,

MP4, M4A etc. which are lossy. This spike generating system is a lossy coding technique as well. This might be very effective in the cases where we do not need all the information from the original sound for a certain ‘purpose’. The purpose of the MP3 or MP4 coded music files are to be played in any sort of musical devices with a good quality of sounds [20] & [21]. Now, human auditory system has its own limitations. So, for example, if the coded sound ignores any frequency content above 20 kHz in the original sound, the reconstructed sound would not make any difference to the listener. The advantages are (in this case) that we need less data to code a sound. Now for some research purpose, lossless coding technique has been developed and they guarantee the same qualities in both original and recoded audio [22]. Now the purpose of this biologically inspired spike based system is to replicate the working procedure inside our cochlea and to understand the intelligibility of different kinds of sounds i.e. amplitude modulation, frequency modulation etc. to segregate the sounds which are coming from different sources. Can we use this spike coding technique to investigate intelligence in sound or to find any pattern in the sound which has similar properties and somehow possibly can recognize it? We human can do it quite easily when we can recognize somebody’s speech voice. And that can be possible for computers or machines to recognize if we can successfully develop an audio coding system which has some similarity with the way our human auditory system works.

1.2 Thesis: Aims and Objective

Coding sound has been an interesting issue for the past few decades. This research concentrates on the decoding of sounds from a spike based representation. The human brain never regenerates the sound from the received spikes from auditory nerves, it interprets those spikes. The reason we have regenerated the sound from its coded state is to see how good the spike code is. If the spike codes are not able to code sound very well, it will not be able to produce good quality of decoded sound. So, the quality of decoded sounds reflects the effectiveness of corresponding spike coding techniques. Here, we have considered three types of spike codes and decoded sounds from them, which are explained next few chapters. The first type of spike coding technique closely follows the way human auditory nerves spike for the vibration of cochlea. So, this spike coding technique has been named as Auditory Nerve (AN) spike codes. That AN spike code

generation system is created by my supervisor Prof. Smith ([3]). Then, the Onset spike codes have been generated which uses those AN spikes to generate onset spikes where there is an increase of energy in the original sound. The third kind of spike coding is Koickal's hardware based spike coding which has been described in literature review (chapter 2).

Both AN and Onset Spike codes have been used to reconstruct the sound from their coded state. The purpose of reconstructing is to see how well those spikes represent sound. So, the reconstruction method is an evaluation of the generated spikes. This re-synthesis of sound will also provide us

- **Effective coding technique:** The comparison between the quality of decoded sounds and their originals can give us the measure of coding efficiency and the effectiveness of the technique.
- **Better compression technique:** The goodness of a compression technique can be assessed by the number of spikes, generated for each sound. AN Spike coding is a lossy coding technique. Onset spikes are more lossy as they are very few compared to the AN. Intelligibility of reconstructed sound from those spikes can measure the quality of compression technique.
- **The intelligibility of spike coding:** As mentioned in the previous section, the spike coding can be more useful in some sorts of environment than others. By decoding the spike codes we can have a measure of idea which type of environments are better or worse for coding spikes.

However, to accomplish all these aims and objectives, we need to be more specific to our aims and our research interest has to be narrowed down to the form of some research questions which can be answered by conducting experiments. They are mentioned in the next section.

1.3 Research Questions

In this thesis, we have set up these following major research questions to meet our aims and objectives mentioned above. These research questions make our aims and objectives more specific. They are as follows

First Question

The three spike codings we have considered are: AN spike coding, Onset spike coding and Koickal's spike coding. As the spike codes represent the sound, it will be possible to decode the sound from its spike coded state. So, how can an algorithm be presented for reconstructing sound from its spike codes and is there any issues which to be considered in case of decoding?

Second Question

How can the AN and Onset spike coding techniques be more efficient to interpret a sound better and generate the spikes quicker?

Third Question

Sound coding is a very common and useful technique today and there are various lossy sound coding techniques like MP3, MP4, WAV etc. Spike coding is another type of lossy sound coding. In general, MP3 and MP4 are quite good at coding sound as the decoded sounds from them are very clear and of good quality. So, the research question that has been raised is which spike coding technique is better to represent a sound?

Fourth Question

There are various types of sounds like: male speech, female speech, choir, speech with background noise as different level of intensity. Which type of sound can be represented the best by a spike code technique?

Fifth Question

Can the spike code be possibly used in speech recognition?

In the rest of this thesis, we have demonstrated how the answers from those questions have been met and the answers are summarized at the conclusion chapter 7.

1.4 Thesis Outline

Firstly, the decoding technique of AN Spike Representation has been invented and it has been described in chapter 3. It has also been well explained that our brain does not reconstruct the sound from the auditory nerve like fiber spikes but the decoding provides

a useful comparison between the original and decoded sound to identify the effectiveness and genuineness of those spike coding techniques. There are many issues which are involved in this decoding technique as described in chapter 3 like: Delay vectors, proper tuning of Gammatone Filterbank number of channels and sensitivity levels and others. The amount of time coding and decoding spikes has also been reduced by inventing a new spike database.

Then the onset spikes generation technique has been investigated by its parameter values as discussed at the beginning of chapter 4. The decoding algorithm has been documented also in chapter 4. There are few other issues, raised at the time of decoding which have been described in the same chapter. Koickal's spike coding technique decodes the spikes after generating them. So, all three different spikes have been finely tuned and then decoded from its coded states. Then the testing has been carried out to evaluate the efficiency and accuracy of those spike codes by comparing decoded and original sounds.

In chapter 5, a subjective test has been carried out. 21 volunteers participated in that sound test of 20 questions with different age, gender and hearing ability. The answers are hypothesized binomially to provide the statistical evidence behind any conclusion. Then an Objective testing has been carried out as mentioned in chapter 6. PESQ (Perceptual Evaluation of Speech Quality) and composite test provides some scores to find out how similar one sound is with another.

Chapter 7 concludes the thesis. It summarizes what has been accomplished in this research project. Then it demonstrates the empirical findings which were obtained throughout this research. Then it answers the research questions mentioned earlier along with its limitations. The direction of future works has also been provided at the end of final chapter Conclusion (chapter 7).

Chapter 2

Literature Review

2.1 Background

This research is based on spike coding techniques for sound; most of them are biologically inspired. As the spike based system is based on human hearing, it is important that we discuss the way our hearing system works. In this background section, we will briefly discuss the basic structure and function of the auditory system, spikes and various spike coding techniques and the background of an exercise to compare the quality of sounds. The auditory system has three major parts according to ([23]). They are:

1. The outer and middle ear
2. The inner ear and basilar membrane
3. The transduction process and the hair cells

The neural responses from the auditory nerves will be discussed in this section as well.

2.1.1 Basic Structure and Functions of The Auditory System

The three major parts of auditory system are described here:

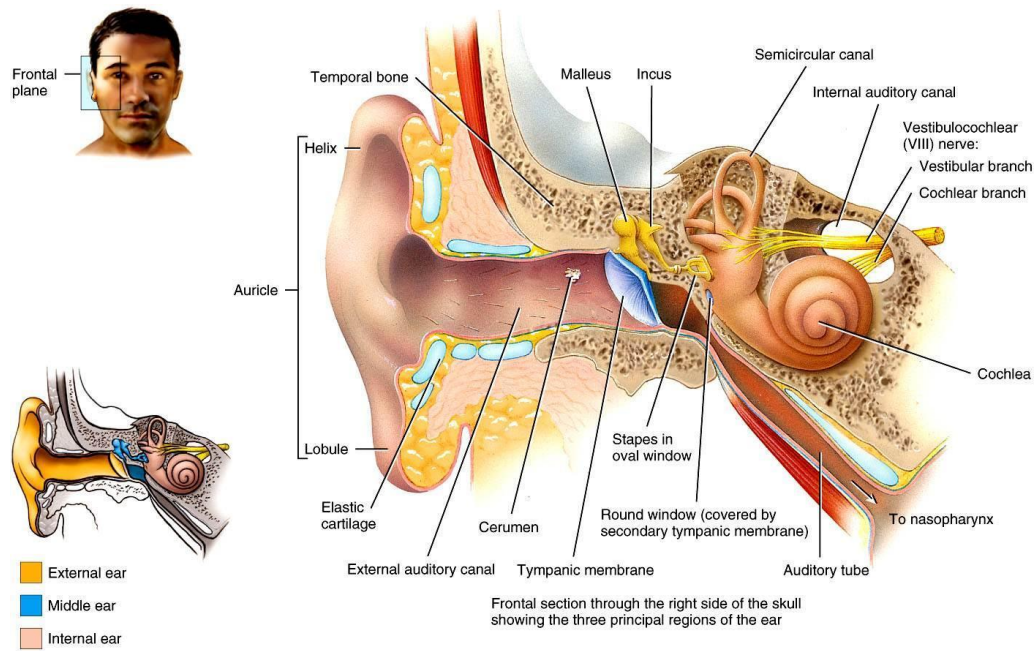


FIGURE 2.1: The Whole structure of Auditory System of Mammals (Humans). The position of Ear and part of the head involved with the sound has been described. The External, Middle and Inner Ear and with all its components have been shown in this detailed picture. Source: [24]

2.1.1.1 The Outer and Middle Ear

This part of the auditory system is similar across most mammals like cats, dogs, rabbits, monkeys etc. [25] pointed out a few animals whose hearing system is very similar to human hearing, however high frequency hearing (above 32 kHz) is common for mammals. Figure 2.1 and 2.2 shows the peripheral part of the human auditory system. The outer ear consists of the pinna and the auditory canal or meatus. The pinna, the visible part of ear, is useful to localize sounds as it modifies the amount of signal reflected down the auditory canal with frequency, depending on the angle between the pinna and the sound source. Sound travels through the auditory canal (see figure 2.1 and 2.2) and vibrates the eardrum or tympanic membrane. The vibration is then carried through the middle ear by three small bones (malleus, incus and stapes), the ossicles, to the cochlea, a bony, spiral shaped structure of inner ear (see figure 2.1 and 2.2). This opening junction is called the oval window. The stapes, the lightest and the smallest bone in the human body, transforms the vibration to the oval window of the cochlea.

The middle ear in mammals is responsible for ensuring sound transform from the outer

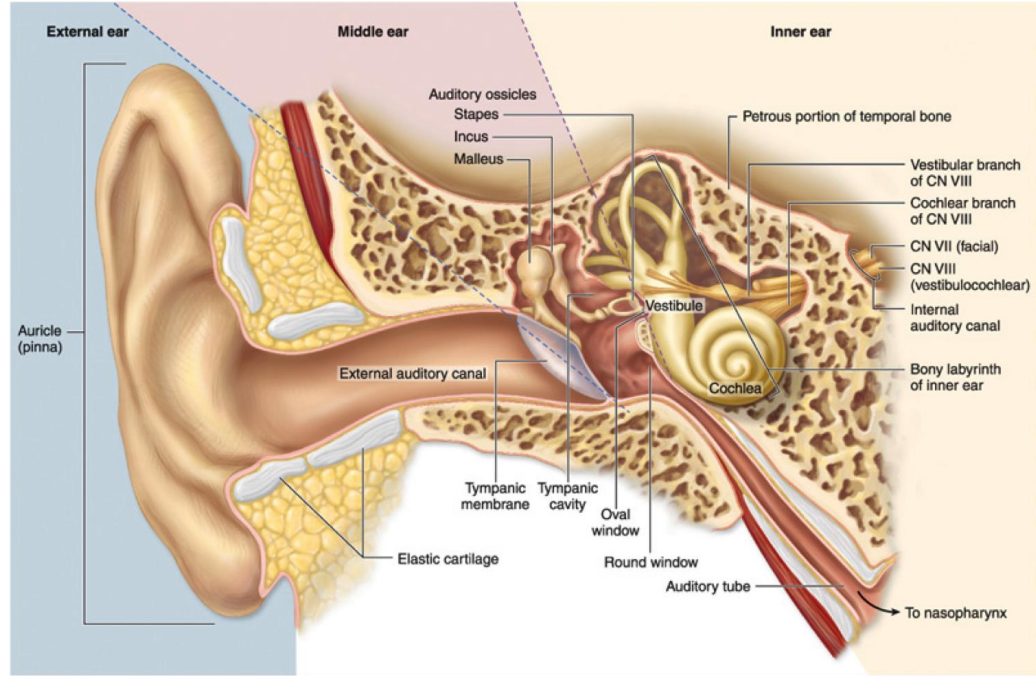


FIGURE 2.2: A cut-away view of the early auditory system showing the parts which is very similar like human. Source: [26]

ear to the cochlea fluid. The middle ear works as a sound transformer and a great reducer of reflected sound. The sound just cannot enter from the air to the cochlea because the most of it will be reflected back. This happens because the acoustical impedance of the oval window to movement is very different from air. Sound between 500 Hz to 4000 Hz is transferred most effectively into the cochlea. This happens because of the difference in effective areas in eardrum and oval window and the lever-like actions of the ossicles ([23]). Another use of middle ear was suggested by [23] is reducing the bone-conducted sounds such as chewing, blood flow. Otherwise these sounds would appear to be very loud which would make distinguishing external sounds from the internal sounds very difficult.

2.1.1.2 The Inner Ear and the Basilar Membrane

Cochlea function provides insight into the auditory perception. The cochlea looks like a spiral shell and it is shaped as spiral. It is filled with incompressible fluids. Two major parts of it are Reissner's membrane and Basilar membrane (BM). The response of BM according to different types of sound will be in our focus in this research project. One

Traveling wave

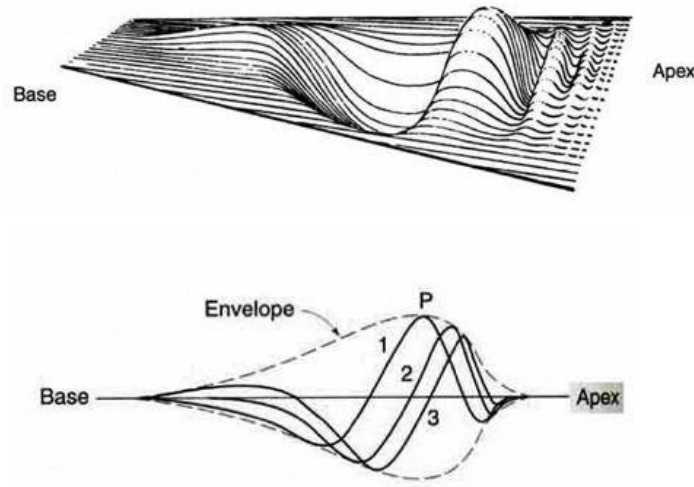


FIGURE 2.3: The displacement pattern moves from left to right denoting rapid decay after the point of maximal displacement. The dotted line represents the envelop made up by cochlea models invented by von Békésy. Source: [27]

end of the cochlea, near oval and round windows, is called the base and the other end is called the apex. At the apex, a small opening, the helicotrema, connects the BM and the walls of the cochlea. This leads to two other outer chambers in the cochlea called scala vestibuli and scala tympani.

When the oval window moves due to the displacement of the stapes, a pressure difference is created across the BM. This causes BM to move in a complex way which varies for different pattern of sounds and over the length of the cochlea. The BM movement takes place like a sinusoid which increases at first but abruptly decreases. Figure 2.3 and 2.4 explains this displacement in detail. The structure of the cochlea also causes this kind of movement. At the base, it is quite narrow and stiff but at the apex it is wider and less stiff. So, the higher frequency sounds create maximum displacement of the BM near the base, whereas they have very little impact on the rest of the membrane. Lower frequency sounds produce a movement which is carried along the BM and reaches its maximum intensity at the apex. Figure 2.3 and 2.4 explain this in detail.

The location of the maximal displacement varies as the BM varies with the frequency

of the incoming sound. Low frequencies result in maximal displacement at the apex and higher frequencies result in maximal displacement near the base as shown in figure 2.4. There is a sense in which this is like a Fourier Analyzer, transforming the signal to the frequency domain but keeping timing information as well. Each point on BM vibrates approximately like sinusoidal manner with the frequency equal to the input sound waveform. Usually the input sound is wideband where many frequencies are concentrated together. Different parts of BM moves as according to those many different frequencies. The frequency for which BM responses the most at a particular point is known as the characteristics frequency of that place on BM. For example, if a 500 Hz sinusoid is applied, each point on the BM moves at that frequency, however one place on the BM vibrates the most. Figure 2.3 and 2.4 illustrates this fact.

The BM movement can be different varying in phase of the vibration but be the same for the frequency of the vibration. Each single point along the BM can be considered as a bandpass filter with a center frequency. The bandpass filter which has been used in this research to generate auditory nerve like spikes is Gammatone Filterbank (GMF) [mentioned in section 2.2]. This is explained later in this chapter. Much research has been undertaken on the BM vibration, as this provides a deep conceptual idea about the relationship between the BM responses and neural responses. This work was initially started by Von Békésy ([28]). He showed that BM is tuned but his work involved only dead animals. BM tuning is different between live and dead cochlea. However other recent research works has revealed that BM is actually more sharply tuned than claimed by Von Békésy as they used the live cochlea in live mammals ([29]). In [29], James Pickles showed in two comparing figures that the responses from a dead cochlea does declines smoothly towards the apex of the cochlea and the real part of impedance is always positive, i.e. cochlea always absorbs energy. However for the living cochlea, the power flux increases rapidly to the peak and drops down very sharply. The real part of the impedance shows that it can be negative which means that the cochlea is amplifying energy there. In other words, energy has been introduced into the traveling wave. Von Békésy (1942) started the work and in time the work has been expanded into specific measurements in traveling wave in 1986 ([30]). In 1997 the responses from BM to the tones at the base of Chinchilla cochlea has been investigated ([31]) and it has been found that ‘responses to low-level (<10 – 20 dB SPL) characteristic-frequency (CF) tones (9–10 kHz) grow linearly with stimulus intensity and exhibit gains of 66–76

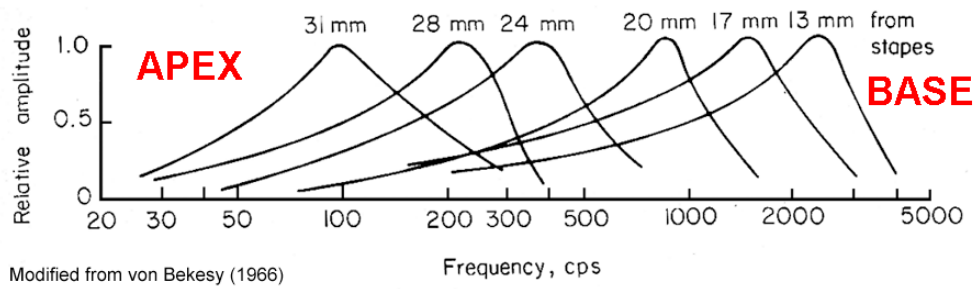


FIGURE 2.4: The instantaneous displacement of BM with six different input frequency. The maximum amplitude has been recorded as 1 and the displacement from the stapes has been measured in millimeter. The displacements have been mentioned with respect to APEX and BASE. Source: [33]

dB relative to stapes motion. At higher levels, CF responses grow monotonically at compressive rates, with input–output slopes as low as 0.2 dB/dB in the intensity range 40–80 dB. Compressive growth, which is significantly correlated with response sensitivity, is evident even at stimulus levels higher than 100 dB. Responses become rapidly linear as stimulus frequency departs from CF. As a result, at stimulus levels >80 dB the largest responses are elicited by tones with frequency about 0.4–0.5 octave below CF. For stimulus frequencies well above CF, responses stop decreasing with increasing frequency: a plateau is reached. The compressive growth of responses to tones with frequency near CF is accompanied by intensity-dependent phase shifts. Death abolishes all nonlinearities, reduces sensitivity at CF by as much as 60–81 dB, and causes a relative phase lead at CF' [31].

Then in 2010, the responses from BM has been studied again and this time it was for the white noise and pure tones ([32]).

In [23], Moore also shows that these sharpening of the tuning largely depend on the physiological condition of the animal or mammal. The better the condition is, sharper the tuning of BM is ([23]).

2.1.1.3 The Transduction Process and the Hair Cells

The hair cells are the part of Organ of Corti and situated between the BM and the tectorial membranes. The hair cells which are the closest to the outside of cochlea are

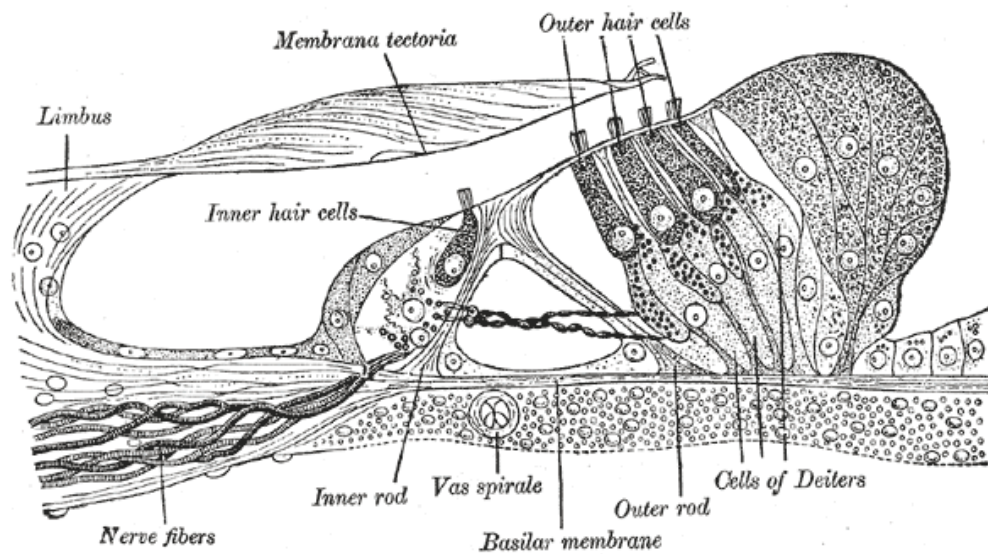


FIGURE 2.5: The cross section of Cochlea showing the BM, the tectorial membrane and Organ of Corti. Source: [34]

called outer hair cells. They make up usually three rows for cats [23] and up to five rows for human (as shown in figure 2.5) and there are about 12000 of them, each is connected with 140 hairs or stereocilia. So, when the BM vibrates or moves, a shearing motion is created between the BM and tectorial membrane. So, the hair cells are displaced as well following that pattern. Figure 2.5 and 2.6 explains the position of hair cells in the cochlea.

2.1.2 Neural Responses and Firing Rates in the Auditory Nerve

There are approximately 30,000 axons in each auditory nerve which carries information from the cochlea to the central nervous system. The auditory nerve carries the information from the cochlea to the brain. They go to the cochlear nucleus in the brainstem. Studies have been conducted where very fine tipped electrodes were used to identify the activities in the auditory nerves [23]. These micro-electrodes have shown the spontaneous firing rate (i.e. firing rate in silence) has been discovered to be between 0 to 150 per second. It has also been shown that the fibers responded better to some frequencies than others. These fibers also show the property of phase locking. Neural spikes tend to occur at a particular phase of the input sound signal.

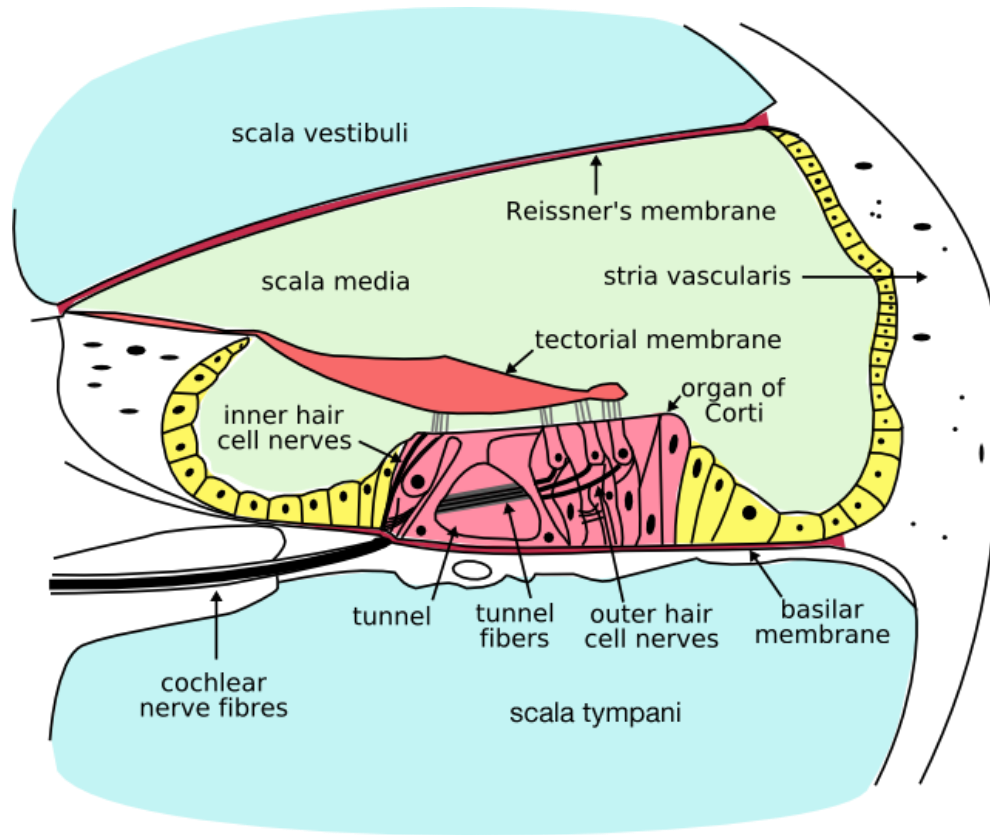


FIGURE 2.6: This is the cross-section of Cochlea where different parts are discussed in the text. Source: [35]

There are two types of auditory nerve cells: Type I and Type II. Type I neurons make up the most of (about 90-95%) the neurons and innervate the inner hair cells. Their diameters are relatively large and they are bipolar. Each type I nerve fiber is connected to only one inner hair cell but an inner hair cell can be connected with up to 30 of type I neurons. Type II neurons make up the rest (5-10%) of the neurons. They are connected to the outer hair cells. In contrast with the type I neurons, type II neurons have relatively small diameters and they are unipolar. Figure 2.7 illustrates this in detail.

Almost 61% of all those nerve fibers have higher firing rate like 18 - 250 spikes per second. 23% of them have medium rates i.e. 0.5 to 18 spikes per second. The rest 16% respond very slowly with the input sound, even less than 0.5 spikes per second ([23]).

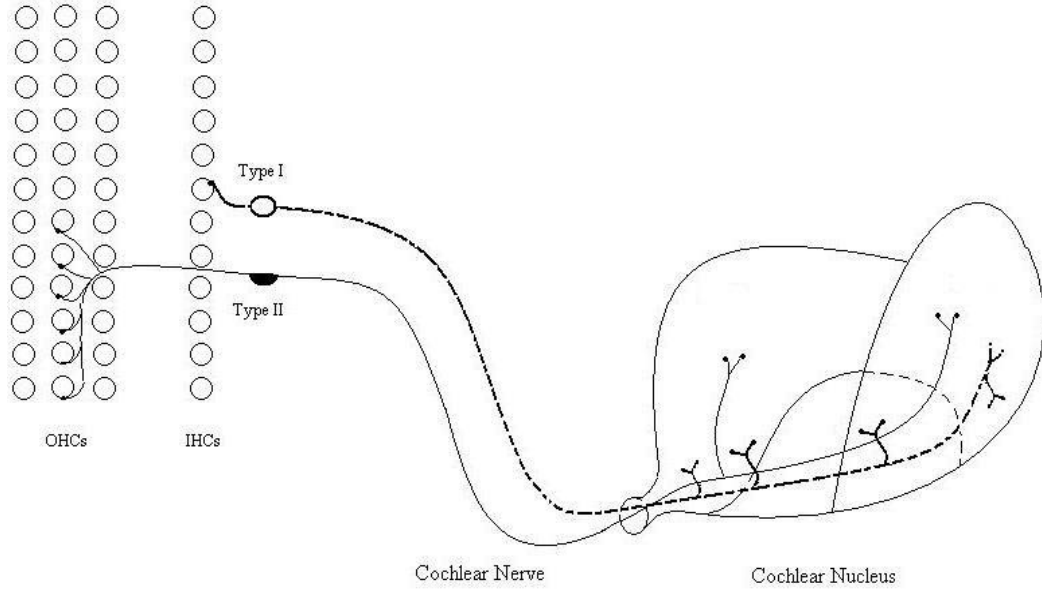


FIGURE 2.7: **Type I and Type II nerve fibers:** The functions of these two neurons has been shown here. Type I fibers (make up about 90-95% of neurons) are connected with inner hair cells and Type II fibers (make up about 5-10% of neurons) are connected with the outer hair cells. Source: [36]

2.2 The Gammatone Filterbank

We use the Gammatone Filterbank (GMF) to evaluate the early auditory system. This filterbank has been used to generate the AN spikes and onset spikes. Paterson [37] worked with complex sounds and showed that gammatone function of order 4 fits very well to the human auditory filters. GMF is very commonly used to simulate the motion of BM. Like the BM, the filterbank has a range of center frequencies. For each center frequency, the filterbank outputs signal from the input sound.

The GMF is described in ([38]). In 1968, de Boer and Kuyper [39] invented an identification method which they named as Triggered Correlation, which was later known as Reverse Correlation. This calculates the impulse response of an auditory nerve fiber. The process of calculating the response is based on the idea that the auditory nerve fiber's discharge is correlated with the input stimulus. 10 years later de Boer and de Jongh [40] developed a mathematical model and the mathematical expression of this method can be given as:

$$g(t) = t^{n-1} e^{-bt} \cos(\omega t) u(t) \quad (2.1)$$

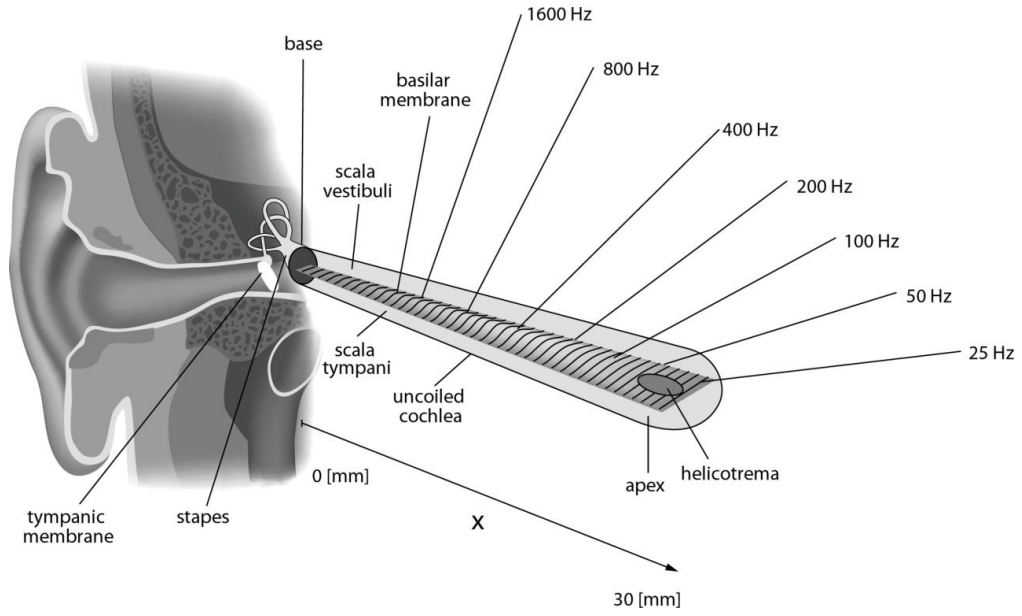


FIGURE 2.8: The uncoiled BM, showing the length and the most receptive frequencies in different places on it.

Source: [41]

where $g(t)$ is the gammatone filter output, n is the filter order, b is related to bandwidth, ω is the radian center frequency and $u(t)$ is the unit-step sequence (i.e. $u(t) = 1$ for positive t , otherwise 0).

de Boer and de Jongh ([40]) worked on two major properties of auditory nerve, frequency-selectivity and partial-synchrony (time locking) between stimulus and response. They suggested a parameter extracting technique which can be used to extract the dynamic response characteristics from an input signal to a system. This technique has been applied to the cochlear physiology and is known as the reverse correlation function. For each fiber, the output signal is a function of time.

The GMF is a model to estimate the BM movement. In December 1987, Paterson and Smith published their work on an efficient auditory filterbank by using GMF ([44]). The next year they summarized some useful properties of GMF ([45]). Later on in 2007, some practical continuous-time filter transfer functions which are quite similar to GMF have been described by Katsiamis, Drakakis and Lyon [46]. This filterbank provides a bridge between auditory physiology and auditory psychophysics based on the frequency selectivity. The data comes from the auditory nerve rather than the BM and sharpness of tuning has been measured at the nerve level.

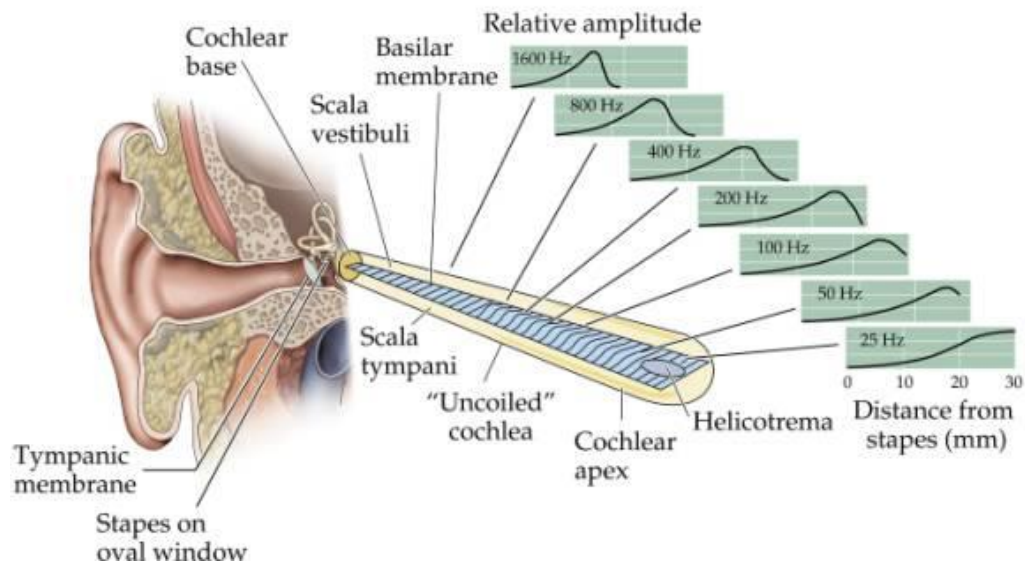


FIGURE 2.9: The uncoiled BM similar to figure 2.8. Showing the frequency plot of the most receptive input-sounds. Source: [42]

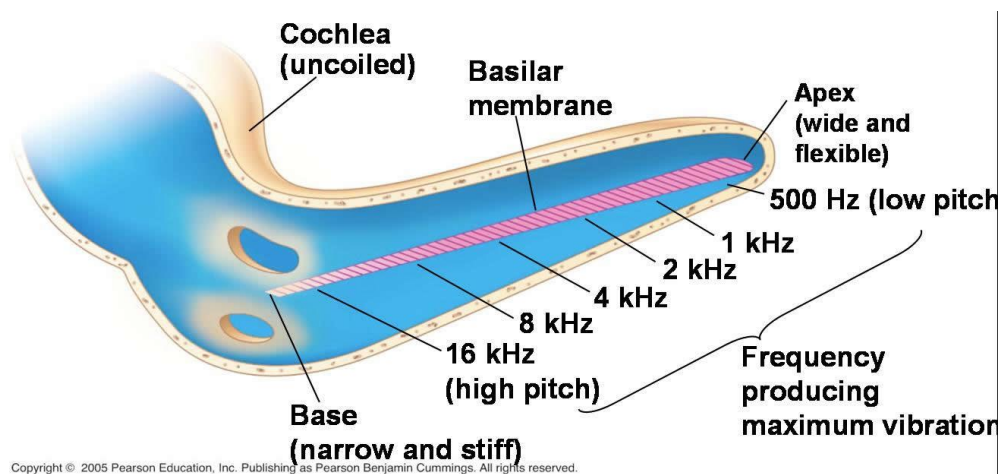


FIGURE 2.10: The uncoiled BM with its frequencies producing maximum vibration. Source: [43]

A few digital approximations have been implemented on the GMF to make it more efficient towards a system. Some of them are:

- the center frequency of the filter (ω) corresponds the frequency shift in the input data.
- the low-pass filter has been applied, where $g_{lp}(t) = t^{n-1}e^{-bt}$
- frequency can be shifted back to the center frequency zone

This makes the equation 2.1

$$g_{lp}(t) = t^{n-1}e^{-bt}u(t) \quad (2.2)$$

Pole-mapping, Bilinear Transform and Impulse invariant Transform are the three digital transformation techniques which are often used.

We have used the GMF which is used to generate AN spikes and onset spikes. Then this filterbank is used to decode those spikes as well. The cut-off frequencies of GMF have also been tuned according to various sampling rate with the maximal frequency set to $<0.5f_s$. That has been demonstrated at chapter 3.

There are other resynthesis works which have been carried out recently from the filterbanks. [17] have implemented Lyon's auditory model and studied Lyon's cochlea models. They have decoded the sounds by using their new correlogram representation algorithm for speech enhancement and sound separation. Similarly [47] developed cochleagrams and correlograms by using a convex projection framework. This estimates a waveform by generating correlogram. Then they can estimate the cochlear channel output by using their spectrogram inversion technique, converting each row of the correlation into a short-time power spectrum. In figure 7 of ([47]), they have shown the complete reconstruction with and without cochleagrams and correlograms of a single sound.

Ewen N MacDonald and others [48] have developed some sets of envelopes from the filterbank output. They have proposed a general optimization approach minimizing the distance between a target envelop representation and that of a reconstructed time-domain signal. However by being inspired of human perception of loudness, a modification of this framework has been proposed in ([48]), which is more accurate inversion of traditional spectrograms. They have reconstructed sound from IHC (Inner Hair Cell

model) inspired envelop extraction. They have used GMF, IHC envelop extraction and went on to directly reconstruct the sound. However afterwards they have proposed their two-step reconstruction work, which has many more advantages over the direct reconstruction. They have also discussed their reconstruction from auditory spectrograms and advised some implications of current IHC models. Inner hair cell model is a part of early auditory processing and adaptation in a revised inner-hair cell model has been proposed by [49].

These works on the resynthesis of sound directly from filterbanks are very relevant and directly connected to this work. They are different than ours as they have analyzed the filterbanks but we have regenerated sounds from our spike codes. They have used the filterbanks which we have used as well. Our filterbank was GMF ([44]) which has been used by ([48]).

2.3 Spikes

2.3.1 What is a Spike?

The Oxford Dictionary definition of a spike is “a sharp increase in the magnitude or concentration of something” [1]. Because it is localized in time, a spike is useful to describe an event. An event is when something ‘happens or takes place, especially one of importance’ [2]. A spike marks a point in time when an event has occurred. Biologically speaking, when sound enters through our ear, the neurons respond to the vibrations of BM as described earlier. This is coded by neuron firing events. Each of those events is coded as a spike, resulting from the sound signal. In this way, the spikes code the phase, loudness and frequency of the signal.

Spike based coding is not new and there are a number of spike codes used in this research area. For example, Koickal [5] has developed a spike based coding system. Koickal has used the spikes which are generated based on transmitting analog signal information. Three different spike coding technique are described later in this chapter.

The comparison between reconstructed sounds from spike based system are discussed later part of this report. For biologically inspired spikes, certain number of channels (for frequency information) and sensitivity levels (for amplitude information) [initially

50 channels and 16 sensitivity levels] have been used in the coding technique. A GMF ([50]) has been used to represent cochlea in the spike generating code. The GMF filters the original signal into each channel frequency. Those channel signals are passed through the spike generating system and for each positive zero crossing in signal, a spike occurs. The RMS (Root Mean Square) value of the previous quarter cycle (where a cycle = $\frac{1}{f_i}$, where f_i is the center frequency of that channel) of that spike has been considered and if that is higher than a certain threshold value, sensitivity levels are assigned for each occurrence of spike.

Our research has been concentrated into the biologically inspired spike based technique here. Cochlea is very sensitive to different frequencies. The range of the highest and lowest frequencies is about 100 Hz to 15 kHz for a middle aged man. The vibration of cochlea can be represented by events and those events are transferred to the inner hair cell followed by the auditory nerve. Auditory nerve carries those signals to brain and then brain interprets those signal and we can hear the sound. Now in the first part, where the sound passed through the cochlea and the inner hair cell sends the events to the auditory nerve, the working strategy has been implemented as spike based coding technique. So, the spike code is similar to the signals which auditory nerve sends to our brain. So the rest of the work is how our brain interprets those event based signals. We note that our brain does not reconstruct the sounds, it interprets it. The reason behind this reconstruction work is to see how appropriate and successful it can be if we reconstruct the sound from the spike code. It also claims that if a reasonable good quality of sound cannot be reconstructed from the spike code, the spike code is not simply effective enough to recognize the intelligence or patterns in the sound.

2.3.2 AN Spike Code

A spike is used to represent an event. As mentioned earlier this spike coding technique is biologically inspired and the cochlea responses are similar to events, they can be represented by spikes.

This AN spike based coding technique in ([3]) and ([4]) is used for the interpretation of sounds. The sound is initially pre-processed by a GMF, and then recoded using spikes (events) in each channel. These events are phase-locked to positive-going zero-crossings, maintaining precise timing (see figure 2.11). Initially, 16 spike trains are used

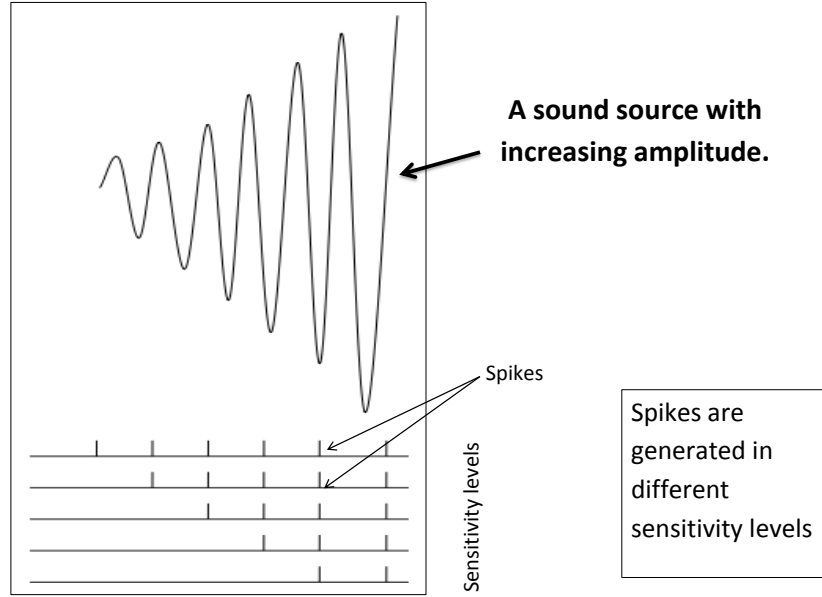


FIGURE 2.11: The spike generation at a glance: In this biologically inspired spike based system, the spikes are generated for each positive-going zero crossing of the input signal. It should also be noticed that the spikes are generated for different sensitivity levels depending on the amplitude of the signal. So, for the high amplitude, which is LOUD, the spikes occur at all the sensitivity levels, but for the low amplitude, which is QUIET, the spikes occur for low sensitivity levels.

per channel, enabling a reasonable dynamic range to be encoded. The coding would be highly suitable for an AER in neuromorphic systems ([51]).

Resynthesis works channel by channel, but needs to avoid inserting additional frequency components not present in the initial signal. This technique tends to produce too many events at higher frequencies as for higher frequencies; the cycle time is very small causing far more positive zero crossings, producing many spikes. The coding-decoding technique can be represented by this $S(t) \rightarrow \text{Spike Trains} \rightarrow S^R(t)$. This coding system splits the original sound signal into different channel signals and then constructs the spike trains for each channel. The number of spike train is $(N \times \xi)$, where N is the number of used channels and ξ is the number of sensitivity levels.

There are some adjustments and compressions done in lossy coding system for example to minimize the silence, setting threshold levels for considering silence and too loud etc. They use different methods to code the sound in a preferable format like mp3, wav.

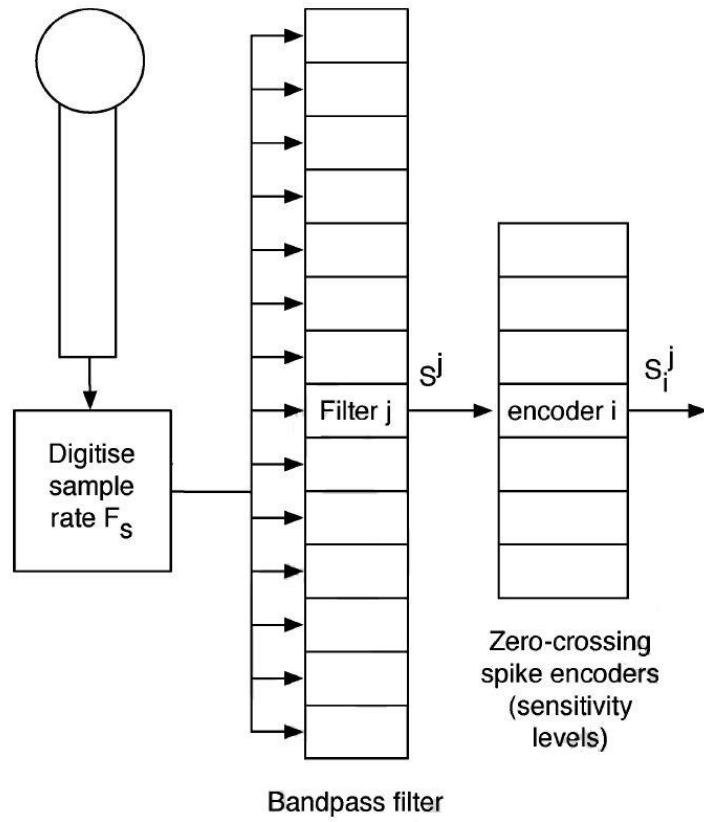


FIGURE 2.12: **AN Bandpass Filter:** The microphone signal is taken at a sampling rate of F_s and passed on to this bandpass filter. There are j numbers of filters and the output from each bandpass filters S^j for band j is passed on to the spike coder. For each positive zero crossing, a spike is generated as S_i^j , where i is the number of sensitivity level. This figure has been shown in ([4]).

Masking is a popular method to compress the sound data. It sets region of acceptable sound zone for human hearing i.e. too loud or quiet and it has capacity to hide noise and any other sound from the original sound. But still those methods are complicated and even more complicated when using many channels. Dr. Koickal [5] has created a spike event coding system to code and decode the sound files. This has been explained later in section 2.3.4.

The reconstruction system decodes the biologically inspired spike based code for any sound file ([3]). It is the inverse technique to code the sound file by spike based coding system. But this technique considers several other facts important towards human auditory systems. In the reconstruction work, the original sound is passed through a GMF which splits up the sound into several channels, and then these are passed through spike generating system. The center frequencies can be denoted by f_c^i , where $i = 1, 2,$

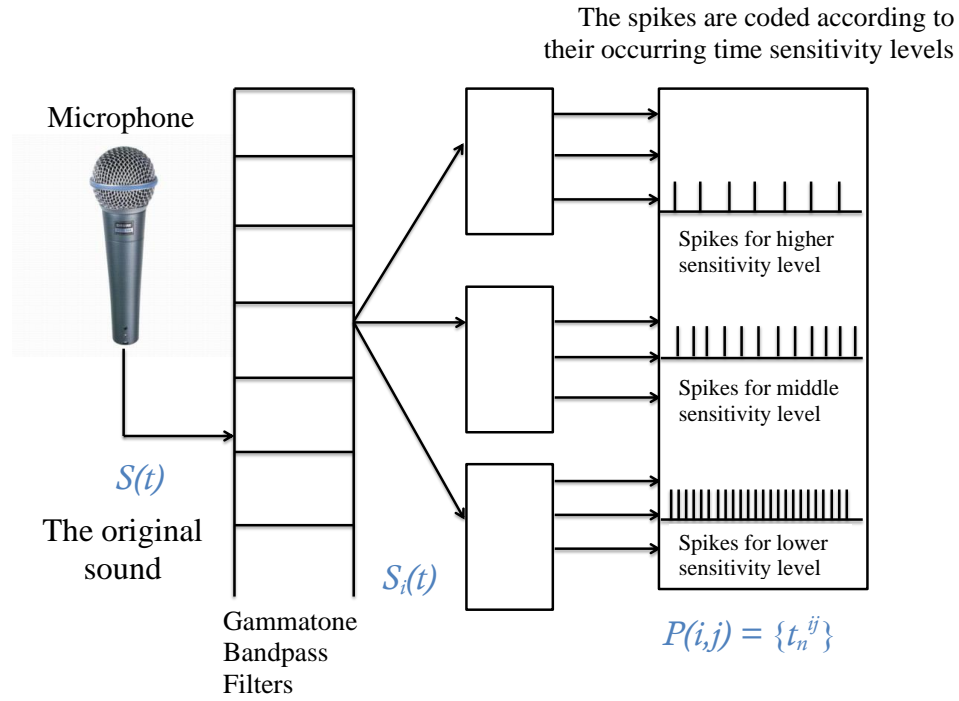


FIGURE 2.13: **The construction of AN spikes from a sound:** The sound file is $S(t)$. It is passed through the GMF and it splits the signal up into N channels & $S(t)$ becomes to $S_i(t)$, where $1 \leq i \leq N$. The sensitivity levels are introduced to record the amplitude of the signal at the occurrence of each spike. Total number of sensitivity levels used in this coding technique is ξ , where $1 \leq j \leq \xi$. Then the spike generation system generates spike trains for each channel and sensitivity level, which are sequence of times. So, $P(i,j) = \{t_n^{ij}\}$.

..., N (where N is the total number of channels used in spike code construction). So, the incoming sound is $S(t)$ [a function of time] and the bandpass filterbank produces $S_i(t)$, where i is the channel number. Then a spike is generated when the value of those bandpassed channel signal each time crosses zero from negative to positive (see figure 2.13).

If only the time values of spike-occurrences are being used here, there would be no information of the amplitude of the sound signal. In the construction work, certain threshold levels and sensitivity multipliers determines the sensitivity level of a spike. Sensitivity levels have a very important role in this sound reconstruction work. A spike is coded by the time value, channel number and the sensitivity level. So, $S_i(t)$ is passed through the spike detecting system and the spikes are coded as $P(i,j) = (t_n^{ij})$ where, $t_1^{ij} < t_2^{ij} \dots < t_{K_i}^{ij}$. K_i is the number of spikes in j th sensitivity level of i th channel. In the work repeated here, $1 \leq i \leq 50$ and $1 \leq j \leq 16$, i.e. we have 50 bandpass channels,

and 16 sensitivity levels. Each $P(i, j)$ is a sequence of spike times. Let also assume that the total number of sensitivity levels used in the spike coding is ξ so that, $j = 1, 2, \dots, \xi$. The threshold levels are a part of the spike code, which calculates the sensitivity level. It can be symbolized mathematically as $\bar{E} = \{E_j\}$, where $1 \leq j \leq \xi$. The equation for the voltage levels is stated in ([3]) as

$$E_{j+1} = D^j E_j \quad \forall j = 0, 1, 2, \dots, 15 \quad (2.3)$$

The value of E_0 is set as 0.0002. Sound power $\propto V^2$, so that if D is set to $\sqrt{2}$ (1.414) for 3dB sound or 2 for 6 dB sound, difference between sensitivity levels. It is also notable that if a spike has occurred in the j th sensitivity level, it has also appeared in all the sensitivity levels j' for $1 \leq j' \leq j$. This has been used in decoding of spike codes in equation 3.4 in chapter 3 (as Θ).

The structure of the AN spikes are as follows:

$$P = P(i, j)|_{i,j} \quad \text{where } i \in 1 \dots N \quad j \in 1 \dots \xi \quad (2.4)$$

where N is the total number of channels and ξ is the total number of sensitivity levels.

So, each P consists of all the spike times in every channel and sensitivity level. For example, if in the coding system, 16 sensitivity levels and 50 channels are used then, in this case, $i = 1, 2, \dots, 50$ and $j = 1, 2, \dots, 16$. The sound-construction data, coded by this spike based coding system, contains the full details of each and every spike, the cochlea center frequency (f_c^i), the time length and sampling rate of the original sound, the number of channels and iterations (sensitivity levels) used to code the original sound and the delay vectors generated inside the GMF for each channel.

2.3.3 Onset Spike Code

What is an onset? The Oxford dictionary definition of onset from the dictionary is ‘the beginning of something’ [52]. In this spike construction work, however, spike represents an event. Previously AN spikes represented a positive zero crossing in the signal. Here, though the onset spikes represents the increase of energy in the signal. The onsets have been used for many different applications like music transcription, sound segmentation, lip synchronization or monaural sound source separation. Onset spikes are a very lossy sound coding techniques, so it must be properly tuned before even the reconstruction work can start. The procedure of choosing the right parameter value to construct onset spikes has been described in chapter 4. The onset detection technique has been described here in detail.

It uses the same band passing system as mentioned in the equation 2.3. Then it starts generating onsets. The neurons in ([3]) are leaky integrate and fire neurons. The generated AN spikes pass through a depressing synapse to those leaky integrate-and-fire neurons or onset neurons. The model has three interconnected populations of neurotransmitter: M, the presynaptic neurotransmitter reservoir (available); C, the amount of neurotransmitter in the synaptic cleft (in use); and R, the amount of neurotransmitter in the process of reuptake (i.e. used, but not yet available again). These parameters are controlled by three different input parameters in the MATLAB code. These parameters are interconnected with each other and each of them contributes to generate the onset spikes. For the reconstruction purpose, these parameter values have been investigated so that the parameter values can be optimized for a certain type of sound. The details about optimizing these values are mentioned at chapter 4 in this thesis.

The rules of interconnections of these parameters are three first order differential equations. They are as follows-

$$\frac{dM}{dt} = \beta R - gM \quad (2.5)$$

$$\frac{dC}{dt} = gM - \alpha C \quad (2.6)$$

$$\frac{dR}{dt} = \alpha C - \beta R \quad (2.7)$$

where α and β are rate constants, and g is positive during a spike, and zero otherwise. Initially the values has been set in the onset spike construction work is $g = 1100$, the total

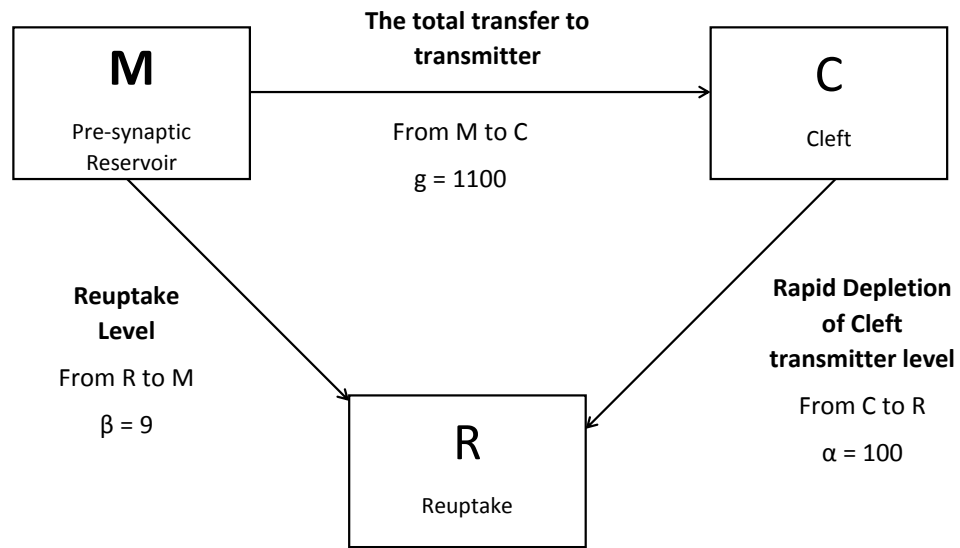


FIGURE 2.14: **The onset generating parameters:** Here are three parameters α , β and g , which are responsible for generating the onset spikes. These parameter's values have been investigated in chapter 4.

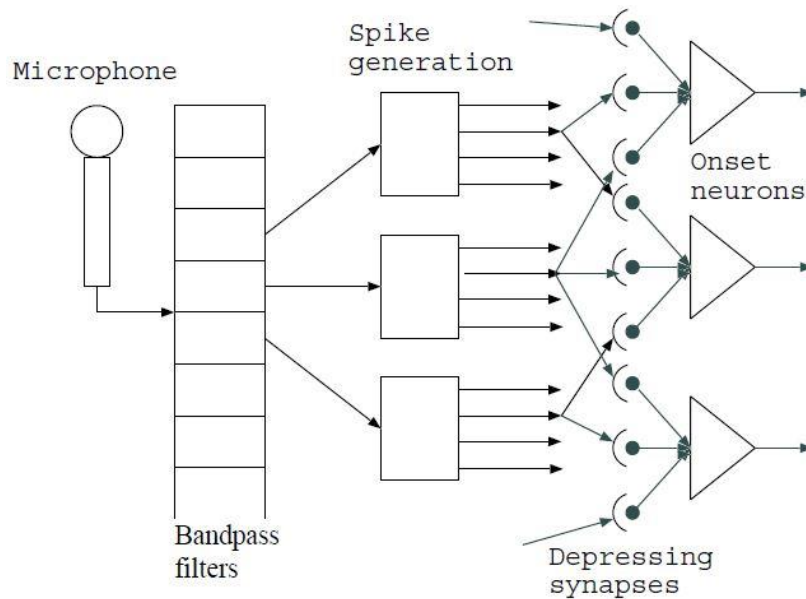


FIGURE 2.15: **The onset generation (source [3]):** The spikes are generated here only for three band and the depressing synapses and onset generation has been shown only for a single level for those three bands.

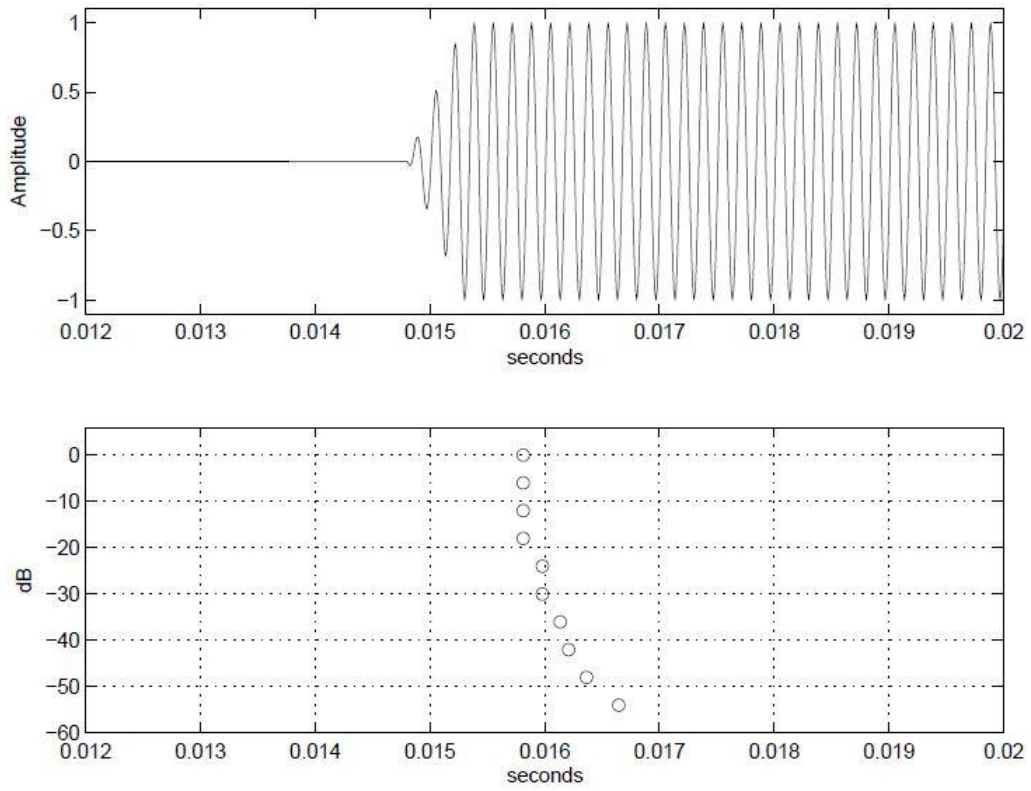


FIGURE 2.16: **The onset spikes are generated from a 6 kHz signal (source: [3]):** The rise time is 0.5 millisecond or 3 cycles. The top figure shows the sound and the bottom one shows the occurrence of onsets as the sound is attenuated in dB, varying in 6 dB steps.

transfer of transmitter from M to C ; $\alpha = 100$ which results rapid depletion of the cleft transmitter level and $\beta = 9$, results slow reuptake of the used transmitter. The other parameters have been: ‘rp_wide (re-uptake parameter)’ = 0.0015, ‘onset_cell_weight (weight of each cell)’ = 1000 and ‘spread_wide (number of AN fibers on each side of the cell)’ = 1. In the case of reconstruction, the values of these three parameters have to be optimized, so that we can get onset spikes at the beginning of each amplitude modulated signal. We have experimentally figured out their optimum values and then we used those values to generate onset spikes. Those onset spikes have been used to reconstruct the sound. The input to the onset neuron for sensitivity level i in bandpass channel b is

$$I_{b,i}(t) = \sum_{j=b-m}^{j=b+m} w C_{j,i}(t) \quad (2.8)$$

where w is the weight of each synapse and $C_{j,i}(t)$ is the neurotransmitter in the cleft associated with the synapse from AN-like fiber from bandpass channel j at the sensitivity

level i . The value m defines the size of the neighborhood innervating the onset neuron.

2.3.4 Koickal's Event Based Spike Code

Koickal [5] has developed a different kind of spike based coding system. There are two types of spikes generated in his code, one is positive going, when the signal change got positive value and another is negatively going, when the signal change have negative value. The output sound signal has to be low passed to get rid of some noise generated in the decoding process.

Koickal's code is relatively easy to implement in hardware devices ([5]), but ignores the possibility of implementation of the biological aspect of human hearing although it can be made adaptive as well. As this research investigates the intelligibility of different kinds of sound and then concentrates to segregate them according to their features, for example- source; something more sophisticated than Koickal's spike based system may be necessary to develop. In [53], Liu has explained the operations of a biological cochlea and introduced circuits which can stimulate those components based on 1D and 2D silicon models. This silicon cochlea designs divided the biological cochlea into several equally-spaced sections and each of them has been modeled by an electronic circuit. Depending on the coupling between these sections, the design has been classified as 1D or 2D. They have admitted that their understanding of it still ongoing. And our biologically inspired spike based coding system follows the similar design. One thing we need to work on in future is to find out the easiest way to implement this technique into micro-electronic chips, as at this point our spike based technique is quite complicated to implement in hardware. However, progress has been made in this area by designing and simulating a silicon cochlea system similar to the real cochlea [54]. The system has biologically faithful frequency response, impulse response, adaption behavior and also suggests that the original sound can be recovered from multiple band-passed channels of spikes.

Koickal et al ([5]) has presented that spike time event coding technique which is useful to transmit the analog signals between configurable analog blocks (CAB). The Configurable Analog Blocks are usually used in Field-Programmable Analog Array (FPAA) which interconnects those analog blocks. Koickal also uses his spike code in Micro-electro-mechanical Microphone project (MEMS). The analog signals from CABs have been

coded as spike trains or spike time events which have been transmitted between CABs (by using AER). They summarize the advantage of using spikes to code sound as:

- The spike representation reduces the power consumption in the circuit and works better in the low energy state.
- Spikes benefit from immunity to metastable behavior, clock skew and low crosstalk.
- The spikes are easy to route for CABs and other multiple chips which provides a bigger scalability.

His spike coder generates a similar signal to the original signal. The generated or feedback signal can be noted as $z(t)$ and the input or original signal can be $x(t)$. The error term $e(t)$ can be identified as the following:

$$e(t) = x(t) - z(t) = x(t) - k_c \int y(t) dt \quad (2.9)$$

where $y(t)$ is the coder output. This output signal is represented by either positive or negative pulses with a short and fixed duration. These are called spikes here. They have been produced by the spike coder and as mentioned before, transmitted between CABs. The tracking step has been noted as δ and it has been produced by each positive or negative going spike.

The spikes have been generated based on the error term $e(t)$ according to equation 2.9. When the error $e > \frac{\Delta e_{th}}{2}$, a negative going spike is generated. On the other hand, if error $e < -\frac{\Delta e_{th}}{2}$; a positive going spike is generated. Otherwise no spikes are generated.

Koickal's code has been created in MATLAB and it produces the spike codes and regenerates the sound from those spike codes as well. The tracking step δ has the default value 5. In Koickal's system, δ has been used to identify the differences between adjacent values in Koickal's system, so as δ increases, there are more spikes. This δ can be used as a input value to the coder. For testing, a 2.055 sec sound file (in the attached DVD, it is at 'Sound Files Under Test/Speech/Test File (My Name)/testfile.wav') has been used. Figure 2.18 and 2.19 has been generated by using the value $\delta = 3$. The number of spikes generated for the value $\delta = 3$ is 70498 spikes, for a 16 bit sound. If the value of δ increases, the number of generated spikes by the spike coder increases as well.

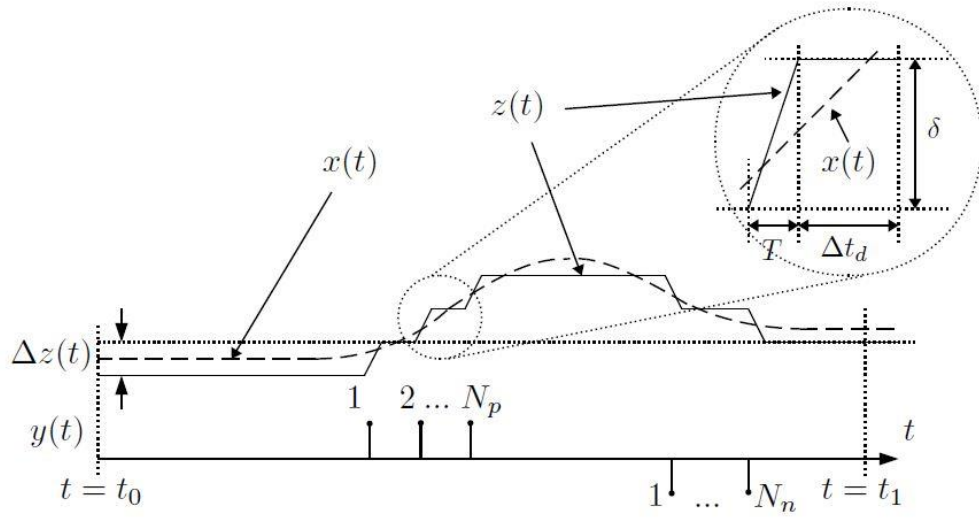


FIGURE 2.17: Koickal's spike coder: Here spike generation has been explained. The time starts from t_0 till t_1 . $z(t)$ is the feedback signal which is forced to track input signal $x(t)$. The bounding error is $e(t)$. $y(t)$ is the coder output. δ is the tracking step and Δt_d is the time interval between two successive spikes. T is the spike 'width'. N_p is the positive going spikes and N_n is the negative going spikes here. Source: [5]

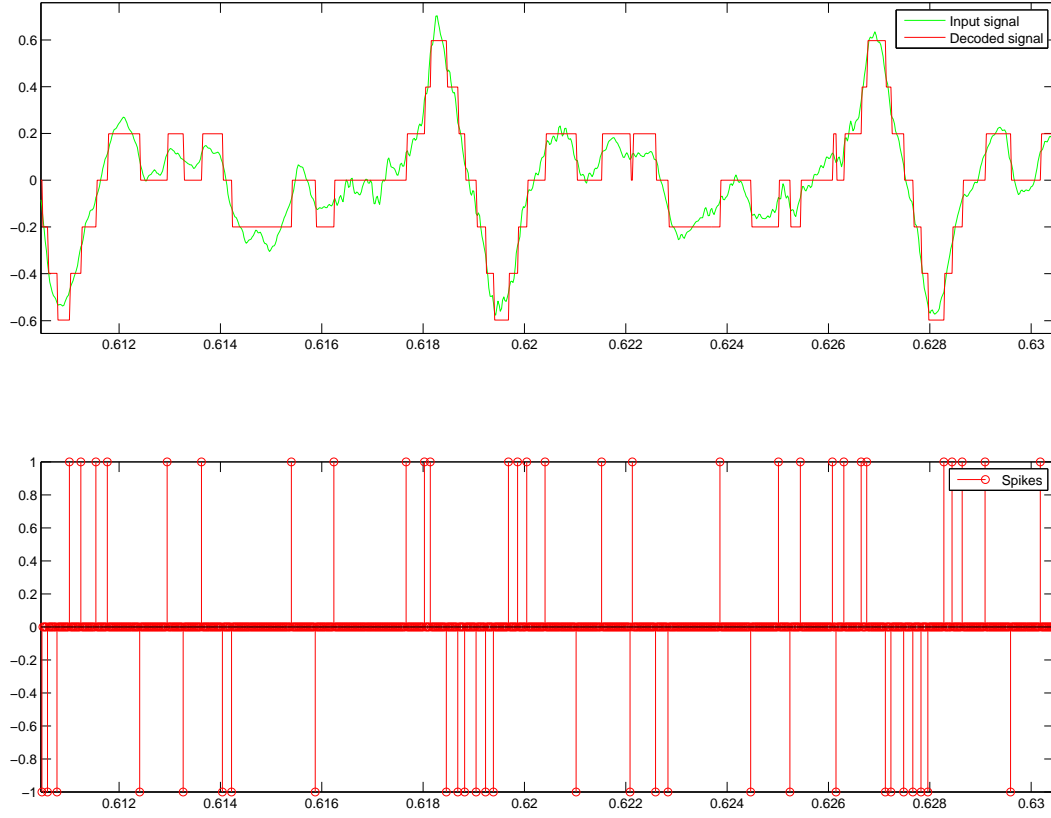


FIGURE 2.18: **Koickal's Spike Coder Output for δ (difference between adjacent values) = 3** : In the top figure, the input signal is a speech (in green) and the decoded signal has been marked in red. Here, the decoded signal follows the original signal quite well. From the bottom figure, the generation of spike codes can be explained. When the decoded signal (in red) having an upward movement, there is a positive spike and for a negative movement, there is a negative spike. The sound is a 16 bit so the range of signal is from $-\frac{2^{16}}{2}$ to $(\frac{2^{16}}{2} - 1)$ i.e. from -32768 to 32767.

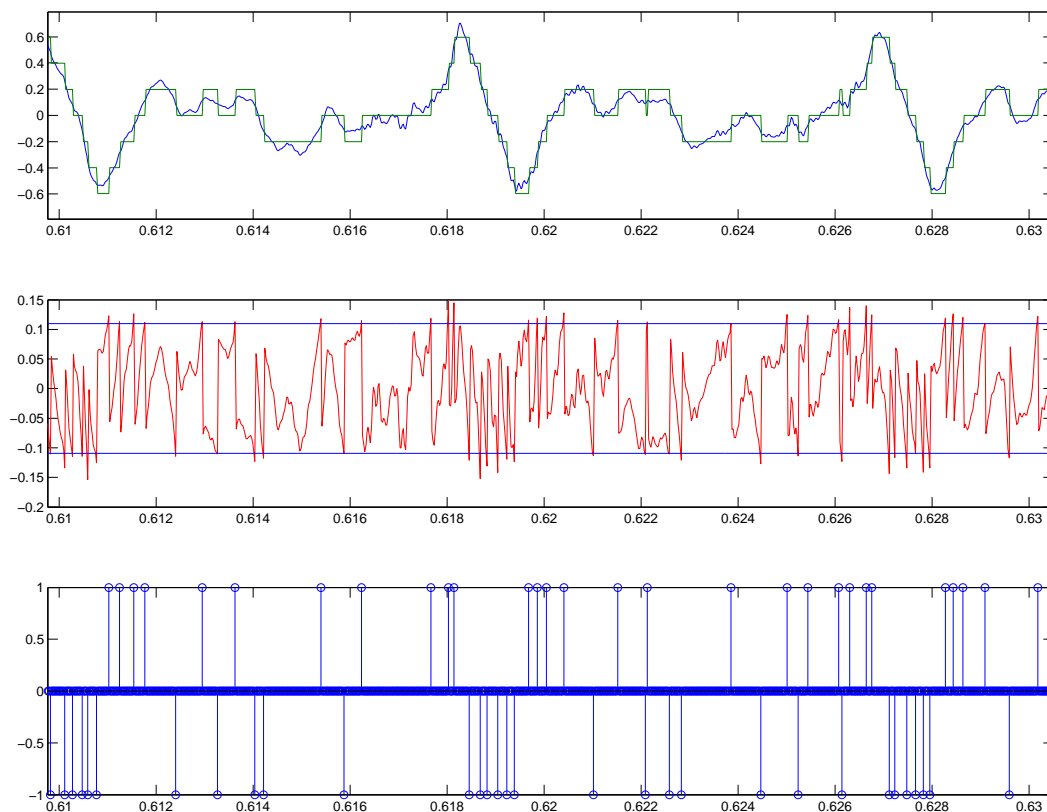


FIGURE 2.19: **Koickal's Spike Coder Output for $\delta = 3$** : Here the error function can also be seen in the middle figure. The very top and the bottom figure duplicates the figure 2.18.

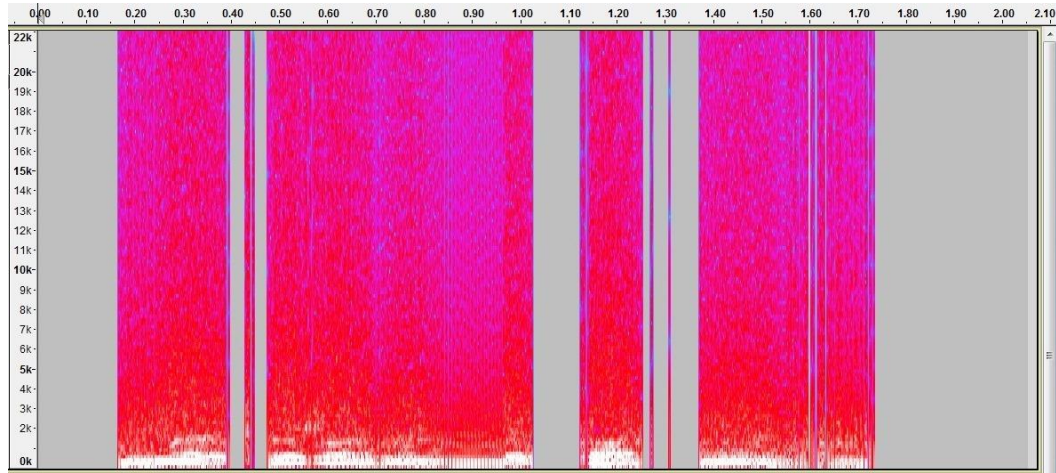


FIGURE 2.20: **Koickal's spike coder output for $\delta = 3$ (without low pass filter)**
: Here few high frequency contents are present. (created in Audacity)

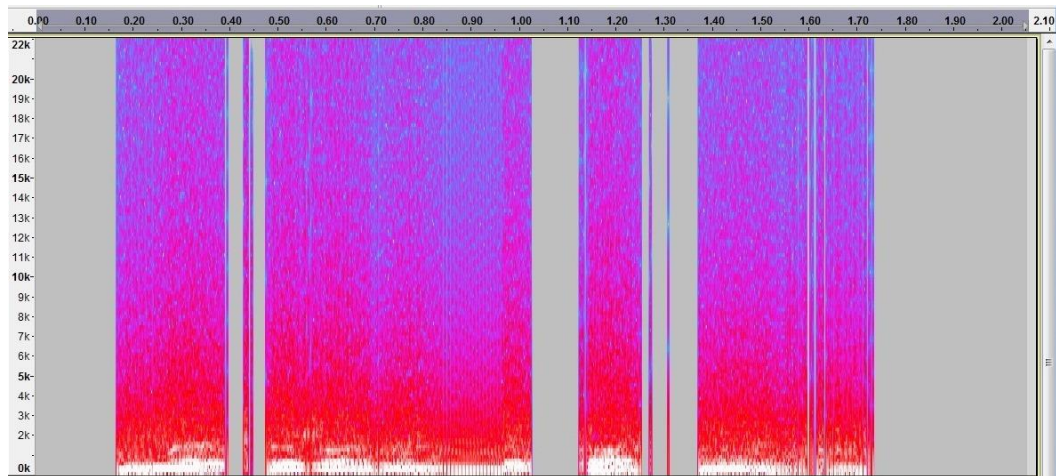


FIGURE 2.21: **Koickal's spike coder output for $\delta = 3$ (low pass filtered at 3000 Hz)**
: There are no frequency content higher than 3000 Hz. This sound signal sounds similar with the previous one without low-passed filtered. (created in Audacity)

In figure 2.20, there are few high frequency contents are present. This is the sound saved in the DVD at 'Sound Files Under Test/Speech/Test File (My Name)/ testfile_NEW_Thomas.3.wav'. This sound has been low-passed filtered at 3000 Hz and the spectrogram has been presented in figure 2.21. That figure 2.21 shows that no frequency contents higher than 3000 Hz are present. This sound has been saved in the DVD at 'Sound Files Under Test/Speech/Test File (My Name)/testfile_NEW_Thomas.3.LPF.wav'.

Those two sounds do not sound any different to each other.

2.4 Other Existing Sound Coding Techniques

2.4.1 MP3

2.4.1.1 MP3 coding technique

MP3 is most commonly used type of compressed music format in the world and very commonly referred to as MPEG-1 or MPEG-2 Audio Layer III since 1991 ([55]). It uses a form of lossy data compression. MP3 lossy coding technique is designed in such a way that it reduces the amount of data required to represent the original sound file, while minimizing distortion. Usually the MP3 files are created with a default setting of bit rate: 128 kilobytes per second. By varying the bit rate, the size of the decoded sound can be varied giving a better or worse sound quality. It should be mentioned here that 128 kilobyte per second is the data rate which is followed by most of compression techniques. At this rate, the reproduced sound is of a good quality and it is difficult to detect the difference between the original and the reproduced sound. If the compression is carried out at any lower than 128 kilobyte per second, then the quality of the reproduced sound drops significantly and if the compression rate any higher than 128 kilobyte per second, the quality of that reproduced sound rises up as well (see [20]).

The compression technique for MP3 is based on the human auditory system. It reduces the accuracy of certain parts of the sound to which our human hearing is less responsive. One example of this is that our human hearing system cannot hear any frequency more than 20 kHz. So, any frequency content over 20 kHz in the original sound is not important for hearing so can be excluded from the code by low-pass filtering it. The compression technique in MP3 is also based on the Perceptual Coding, sometimes called Auditory Masking. Equal Loudness Contour is a part of this process, which takes into account the sound pressure in decibel shown in figure 2.24. Our Ear is most sensitive with the frequency 2 - 4 kHz as mentioned by [20].

2.4.1.2 Comparison with Spike Coding Technique

Likewise our spike code is a lossy coding technique and it does consider the biological parts of human hearing system. The spike code does not consider any frequency content

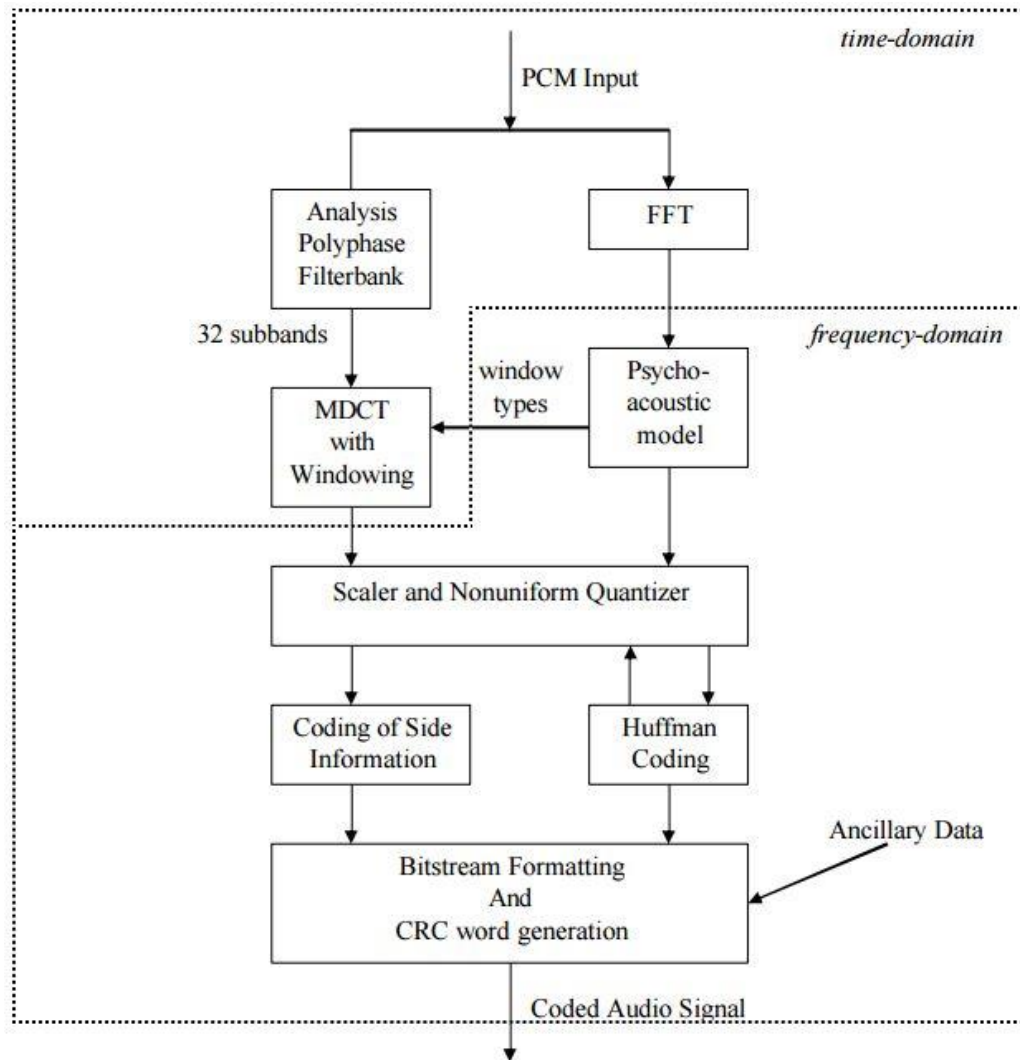


FIGURE 2.22: **MP3 Encoding:** This encoding process is very complex. The simplified version of encoding is shown here. Source: [20, Figure 6.1]

over 20 kHz present in the original sound and so the decoded sound have only the frequency content with less than 20 kHz.

Spike code and the MP3 coding technique are similar in that they both are lossy coding technique. But the major difference is that spikes are generated based on the auditory nerve. It is similar to the representation of spikes which occur in the hair cells connected with the cochlea and brain. The spikes are decoded only to see how well they can be to represent a sound. On the other hand MP3 compression technique decodes the sound so that the size of the coded sound can be less than original but keeping the quality of the sound almost the same. This technique has been adopted as a better compression technique, whereas biologically inspired spikes have been chosen to represent the spikes

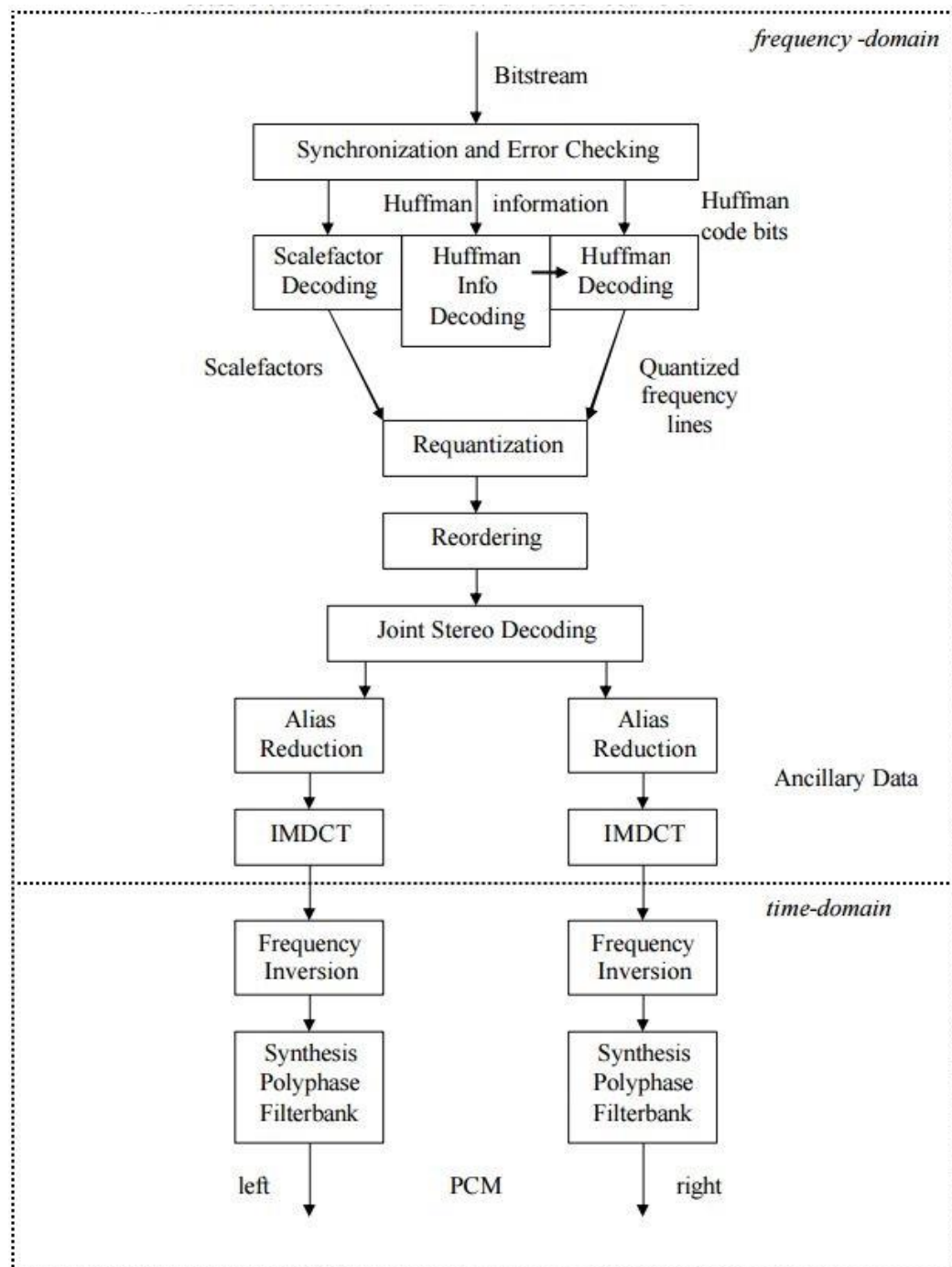


FIGURE 2.23: **MP3 Decoding:** This process is very complex as well like encoding process. This is the simplified version of decoding. Source: [20, Figure 7.1]

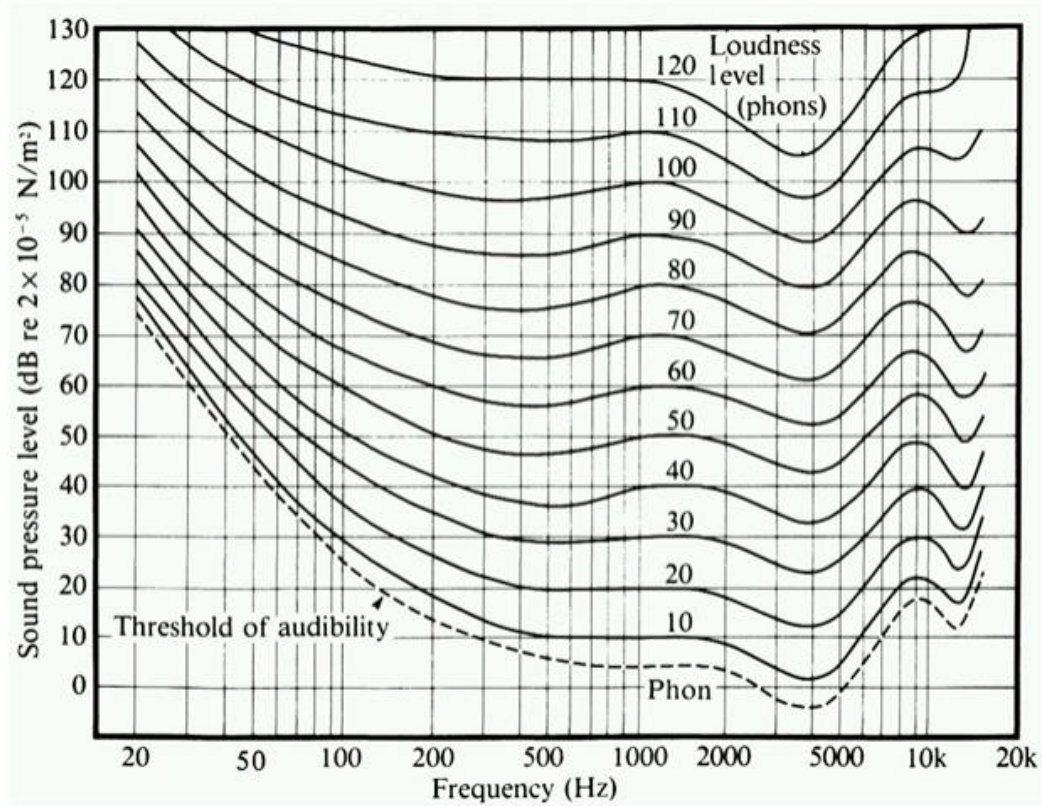


FIGURE 2.24: **An equal-loudness contour:** The peak sensitivity is around 2 - 4 kHz. This is the frequency where the human voice centers the most. Source: [56]

generated inside our BM and cochlea. It has to be admitted that MP3 sound code produces much better quality of sound than our decoded spike codes.

2.4.2 MP4 & AAC

2.4.2.1 MP4 & AAC coding technique

MP4 is not a updated and higher version of MP3 ([57]). It is completely different from it. Unlike MP3, MP4 is able to code audio, video and still images ([58]). The audio files coded by MP4 have the extension '.M4A'. The reason it is called differently from MP4 is that there are two versions of audio coded by MP4, (i) lossy Advanced Audio Coding i.e. AAC and (ii) Apple lossless formats. We will only consider the lossy coding technique here ([59]).

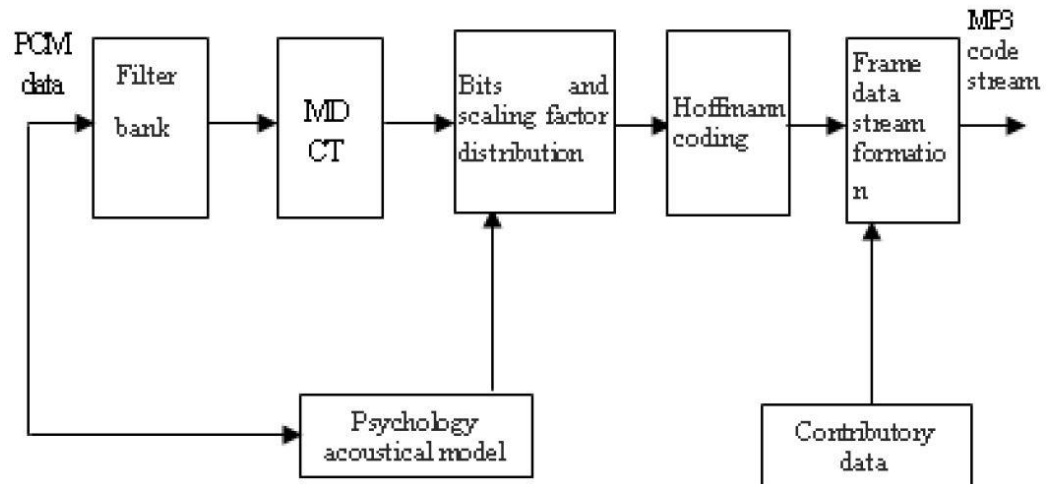


FIGURE 2.25: **MP3 Encoding process:** Source: [60, Figure 1]

The secret of AAC coding's efficiency is that it uses a wideband or HD Voice audio coding algorithm which reduces the amount of data needed to represent high quality audio ([61]). A few major procedures which AAC follows:

- Redundant and irrelevant signal components are discarded.
- The Modified Discrete Cosine Transform (MDCT) provides the transformation from time domain to frequency domain.
- Corrupt samples are prevented by using the Luhn mod N algorithm to each single frame.
- It provides for sampling frequencies between 8 kHz to 96 KHz and 1 to 18 channels.

AAC provides major improvements over MP3. AAC has been designed to be the successor of MP3 ([21]). If the same bit rate is been applied then AAC provides greater flexibility and transparency than MP3 ([12]). However later this has been proved that AAC works better than MP3 for the bit rate of around 100 kilobyte per second. There are certain encoders in MP3 which actually encodes the sound better than AAC. Some improvements of AAC over MP3 are as follows:

- AAC provides a bigger range of sampling frequency (8 to 96 kHz) than MP3 (16 to 48 kHz).

- AAC can support up to 48 channels whereas MP3 can support maximum 5.1 channels in MPEG-2 mode.
- Bit Rates and the variable frame lengths are more flexible in AAC rather than MP3.
- AAC coding is more efficient and accurate than MP3.
- AAC works much better for the stereo type frequency higher than 16 kHz than MP3.

2.4.2.2 Comparison with Spike Coding Technique

Again AAC is a lossy coding technique like the spike coding techniques and AAC produces much better quality of reproduced Audio than our Spike code which will be discussed later in this thesis.

2.4.3 WMA (Windows Media Audio)

2.4.3.1 WMA Coding Technique

WMA audio file format and technology has been developed by Microsoft in 1990s. This is another lossy coding technique. WMA has four different codecs and the most popular one is Windows Media Audio (WMA) ([62]). The first version of WMA was released in 1999 and it was able to encode up to 48 kHz with two discrete channels or stereo type sounds. Later version of WMA used variable bit rate and average bit rate technologies in it. Like AAC, WMA also uses Modified Discrete Cosine Transform (MDCT), which is a little bit different from AAC. WMA competes with Microsoft's other sound codec Speex. It uses both low pass and high pass filtering of sound which lies outside of human hearing frequency range to maximize the output sound quality. Some of Microsoft's audio codecs are capable to distinguish the voice and music automatically.

The sound quality produced by the output signal is better than MP3 as per claimed by Microsoft. Some tests of WMA's sound quality are

- WMA is better than LAME MP3 at 32 kilobytes per second.

- WMA is lower quality than AAC at 80 kilobytes per second.
- At 128 kilobytes per second, WMA is more or less the same with MP3 and AAC.

2.4.3.2 Comparison with Spike Coding Technique

So, WMA is a lossy coding technique which is focused to human voice ([63]). As mentioned earlier, some of the codecs of WMA is able to separate voice with music automatically. However the output signal of spike codes is not as good as WMA, as spikes codes are based on the auditory nerves.

2.5 Background of Subjective Sound Testing

As this research involves the comparison of different spike coding techniques, testing has been conducted to compare them. After that the results has been analyzed statistically and finally the conclusion has been drawn from there.

There are three different types of sounds i.e. male, female and musical types of sounds which are used in this subjective testing as explained in detail in chapter 5.

2.6 Background of Objective Sound Testing

2.6.1 Perceptual Evaluation of Speech Quality Test

The previous sound testing was based on the human-hearing evaluation. Another test has been conducted based on the Perceptual Evaluation of Speech Quality (PESQ) (Source: www.pesq.org). This is a well-known and standard test. This is widely used among the phone manufacturers, network equipment vendors and telecoms operators.

In our sound testing we have used two speech sound file pronounced by a male and a female. So, a regular and a composite PESQ test will be appropriate to compare each of them with their original sound. We have also run this test on other three ‘String’ and ‘Percussion’ types of sounds. This test has been used in an audiovisual speech filtering in noisy environment ([64]). The PESQ parameter values has been described in ([65]) and ([66]).

In [66], Yi Hu and P. Loizou have worked on and improved this PESQ test and other subjective quality evaluations. The quality evaluation tests were first designed according to ITU-T recommendation P.835. This methodology was designed to reduce the listener's uncertainty in a test based on signal or background noise or both. This method helps to achieve the subjective evaluation test reports like below

- the original speech signal based on a five-point scale of signal distortion (SIG). SIG has been detailed in table 2.1 in that subjective test.
- the background noise signal based on a five-point scale of background intrusiveness (BAK). BAK has been detailed in table 2.2 in the subjective test.
- the overall effect using the scale - Mean Opinion Score (OVRL). They can be classified like: [1=bad, 2=poor, 3=fair, 4=good, 5=excellent] in the subjective test.

Scale Points	Remarks
5	Very natural, no degradation
4	Fairly natural, little degradation
3	Somewhat natural, Somewhat degraded
2	Fairly unnatural, fairly degraded
1	Very unnatural, very degraded

TABLE 2.1: Scale of Signal Distortion (SIG). Source: Table 1, [66]

Scale Points	Remarks
5	Not Noticeable
4	Somewhat Noticeable
3	Noticeable, but not intrusive
2	Fairly conspicuous, somewhat intrusive
1	Very conspicuous, very intrusive

TABLE 2.2: Scale of Background Intrusiveness (BAK). Source: Table 1, [66]

There are five widely used objective speech quality measures: segmental SNR (segSNR), weighted-slope spectral (WSS) distance ([67]), perceptual evaluation of speech quality (PESQ) ([68]), log likelihood ratio (LLR) and Itakura-Saito (IS) distance measure ([69]).

Among those above mentioned five measurements, the most sophisticated one is PESQ, which is recommended by the International Telecommunication Union (ITU) for speech quality assessment ([70]). The PESQ final score is computed by a linear combination of the average disturbance value D_{ind} and the average asymmetrical disturbance values A_{ind} . According to ([66]), the formula is:

$$pesq = a_0 - a_1 \times D_{ind} - a_2 \times A_{ind} \quad (2.10)$$

where $a_0 = 4.5$, $a_1 = 0.1$ and $a_2 = 0.0309$. Multiple linear regression analysis has been used to compute the values of parameter: a_0 , a_1 and a_2 . Thus they can obtain the three pesq measures mentioned earlier: $pesq_s$, $pesq_b$ and $pesq_o$ ([66]).

The composite objective measures are calculated by optimizing the use of linear regression analysis. In this work, the PESQ and the composite measures have been calculated to compare the original, AN-Decoded, AN-Onset-Decoded and Koickal-Reconstructed sounds. The composite measures involved three unique evaluation parameters [66]. They are:

1. **C_{sig}** : This is the measure which we have used to estimate the signal distortion (SIG). This has been formed by linearly combining the LLR, PESQ and WSS measurements. The combination is as follows:

$$C_{sig} = 3.093 - 1.029 \times LLR + 0.603 \times PESQ - 0.009 \times WSS \quad (2.11)$$

2. **C_{bak}** : This is the measure which we have used to estimate the noise distortion (BAK). This has been formed by linearly combining the PESQ, WSS and segSNR measurements. The combination is as follows:

$$C_{bak} = 1.634 + 0.478 \times PESQ - 0.007 \times WSS + 0.063 \times segSNR \quad (2.12)$$

3. **C_{ovl}** : This is the measure which we have used to estimate the overall quality. This has been formed by linearly combining the PESQ, LLR and WSS measurements.

The combination is as follows:

$$C_{ovl} = 1.594 + 0.805 \times PESQ - 0.512 \times LLR - 0.007 \times WSS \quad (2.13)$$

Two MATLAB functions have been used to calculate the PESQ, LLR, segSNR and WSS. Based on these measurements, the function also calculates the C_{sig} , C_{bak} and C_{ovl} . The MATLAB function ‘pesq’ and ‘composite’ has been mentioned at Appendix D.

The MATLAB functions can be executed like: ‘[pesq_mos]=pesq(sfreq, cleanfile.wav, enhanced.wav)’, where ‘sfreq’ is the sampling frequency in Hz (8000 or 16000 Hz), ‘cleanfile.wav’ contains the clean speech file and ‘enhanced.wav’ contains the enhanced file. The composite function can be called like: [c,b,o]=composite(‘sp09.wav’, ‘enhanced_logmmse.wav’) where ‘c’ or ‘ C_{sig} ’ is the predicted rating of speech distortion; ‘b’ or ‘ C_{bak} ’ is the predicted rating of background distortion; ‘o’ or ‘ C_{ovl} ’ is the predicted rating of overall quality.

Chapter 3

Reconstructing sound from AN Spikes

3.1 The Purpose of De-coding from Spike Code

We restate that our brain does not reconstruct the sound from the AN spikes; it interprets those spikes. However, by re-creating the sound from its spike based coded state, we can understand how good the spike coding is by comparing the quality of the recreated sounds. Both subjective and objective testing, explained in chapter 5 and chapter 6, have been carried out to compare the qualities of those recreated sounds.

In the previous literature review chapter, we have introduced three different types of spikes: AN spikes, Onset spikes and Koickal's spikes. In this section, we will talk about decoding from AN spikes only, which has been my contribution. The next chapter will discuss onset spikes.

3.2 Resynthesis Algorithm

The reconstruction work aims to generate the sound from AN spike code as mentioned in equation 2.4, $P(i, j)$, where $\{P(i, j) : \text{where, } i = 1, 2, \dots, N \text{ \& } j = 1, 2, \dots, \xi\}$, which consists of all the spikes and their time values according to different channels and different sensitivity levels.

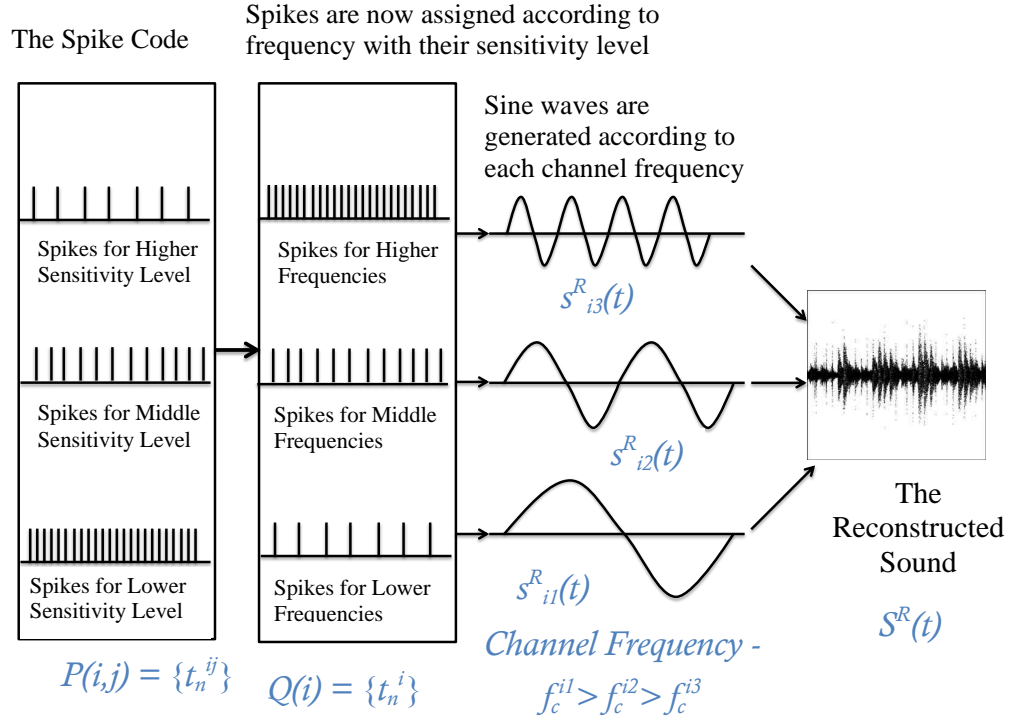


FIGURE 3.1: **The re-construction of sound from AN spike code:-** The spike trains for each channel and sensitivity level are received from the AN spike code, $P(i, j) = \{t_n^{ij}\}$, where i represents channels and j represents sensitivity levels. Then they are assigned according to each channel by combining their sensitivity levels and using the highest sensitivity level. So, $P(i, j) = \{t_n^{ij}\}$, becomes $Q(i) = \{t_n^i\}$, where $t_n^i = (\hat{t}, \hat{j})$ for \hat{t} = time and \hat{j} = occurrence in highest sensitivity level of that spike with occurrences for sensitivity level $1 \leq \hat{j} \leq \xi$ (ξ is the total number of sensitivity levels). The sine waves are created for each channel according to each occurrence of spikes. At the end, the regenerated signals for each channels are summed to get back the reconstructed sound signal $S^R(t)$.

To reconstruct the signal, the first step is to reorganize and assign each spike according to each channel along with an occurrence value which is the highest sensitivity level where that spike has appeared. At this step, the final spike trains can be represented as a two dimensional sequences like $Q(i) = \{t_n^i\}$, where $t_n^i = (\hat{t}, \hat{j})$ for \hat{t} = time and \hat{j} = the highest sensitivity level in which that spike has occurred, $1 \leq \hat{j} \leq \xi$. So for example, t_n^i and t_{n+1}^i are two adjacent spikes with time value $\hat{t}_n^i, \hat{t}_{n+1}^i$ and occurrence in highest sensitivity level $\hat{j}_n^i, \hat{j}_{n+1}^i$ in i th channel. ¹ Figure 3.1 above shows the procedure briefly.

The next step is to compensate for the delays generated inside the GMF. The filterbank delays differ for each channel, so the delay vectors can be represented by D^i , where

¹This step takes significant amount of time to assign the spikes from $P(i, j)$ to $Q(i)$. Later the spike generating system has been amended so that the spikes can be coded in $Q(i)$ format. This improves the processing time of AN spike decoding as the spikes are already coded as $Q(i)$. The processing time improvements have been described later in this chapter.

$i = 1, 2, \dots, N$. The delay is compensated by deducting D^i from the time values \hat{t}_n^i in $Q(i)$. So the delay compensated structure becomes $Q^d(i)$

$$Q^d(i) = \{\hat{t}_n^i - D^i\} \quad \text{for } i = 1, 2, \dots, N \quad (3.1)$$

The delay vectors decrease with the increase of the cochlea center frequencies: here $f_c^1 < f_c^2 < f_c^3 < \dots < f_c^i$ and $D^1 > D^2 > D^3 > \dots > D^i$, where f_c^i is cochlea center frequency and D^i is delay vector. So $D^i \propto \frac{1}{f_c^i}$ (see [38, equation 41, page 69]). The effect of delay vectors is discussed later in this chapter.

$Q^d(i)$ is the sequence of delay compensated time values. At this step, a sine wave ($\hat{s}_k^i(t)$) is generated for each occurrence of a spike. This is the inverse of how the spike was generated. In the AN Spike construction, a zero crossing in the values of the bandpassed sound signal results in a spike. So, for each occurrence of spike, a zero crossing has to be generated. Now at this point, it should be remembered that t_{k-1}^i and t_k^i are two consecutive spikes. A sine wave has been created between two occurrences of spikes. So, the sine wave, s_k^i , is created between the occurrences of time values \hat{t}_{k-1}^i and \hat{t}_k^i of those two spikes. The formula to generate a single cycle of sine wave is

$$\hat{s}_k^i(t) = \sin(2\pi T_k \frac{1}{\hat{t}_k^i - \hat{t}_{k-1}^i} t) \quad (3.2)$$

where,

$$T_k = [\hat{t}_{k-1}^i + \frac{1}{R_{SM}}, \hat{t}_{k-1}^i + \frac{2}{R_{SM}}, \dots, \hat{t}_k^i] \quad \text{for } k = 2, 3, \dots, K^i \quad (3.3)$$

where R_{SM} is the sampling rate of the original sound signal and K^i is the total number of spikes in i th channel.

Now, for each spike, there is an occurrence value attached to it, the highest sensitivity level of occurrence of a spike. That tells how high the amplitude of the signal is between two consecutive spike occurring times. The value of the multiplier J_k^i for each sine wave is

$$J_k^i = \left(\frac{\Theta_{\hat{j}}}{\Theta_{\xi}}\right) \quad \text{for } k = 1, 2, 3, \dots, K^i \quad (3.4)$$

where, Θ is the threshold levels obtained from spike codes. ξ is the number of sensitivity levels used in the spike based code. So, for example, if a spike occurs at sensitivity level 3, the multiplier will be $\frac{1}{0.0362} \times 0.004 = 0.0110$, where 0.0362 is the threshold for the highest sensitivity level 16 and 0.0004 is the threshold for sensitivity level 3.

At this step, $\hat{s}_k^i(t)$ is multiplied by the multiplier J_k^i to get the right amplitude. A ramp technique, which is described later in equation 3.20, has been implemented for the multiplier as an extension of this reconstruction work. In the construction work, for a spike to occur at time \hat{t}_k^i , a high amplitude in the signal is necessary just before the time \hat{t}_k^i (see equation 2.3 at chapter 2). So, the sine wave \hat{s}_k^i should be multiplied by J_k^i , not J_{k+1}^i or J_{k-1}^i . So, the sine waves with right amplitudes ($s_k^i(t)$) are

$$s_k^i(t) = (J_k^i)[\hat{s}_k^i(t)] \quad \text{for } k = 2, 3, \dots, K^i \quad (3.5)$$

So, the sine wave has been generated for all the spikes occurred in each channel. Now for the very first spike, an artificial sine wave has been generated with the length as the time value between the first and second spike in that particular channel. That sine wave is added to the front of the first spike's occurrence. That signal becomes

$$s_1^i(t) = (J_1^i)[\sin(2\pi T_1 \frac{1}{\hat{t}_2^i - \hat{t}_1^i} t)] \quad (3.6)$$

where,

$$T_1 = [(\hat{t}_1^i - T_2) + \frac{1}{R_{SM}}, \hat{t}_{k-1}^i + \frac{2}{R_{SM}}, \dots, \hat{t}_1^i] \quad \text{for } k = 2, 3, \dots, K^i \quad (3.7)$$

To get the reconstructed signal for a single channel, all the $s_k^i(t)$ are concatenated together. It can be expressed in this way

$$s_i^R(t) = \parallel_{k=1}^{K_i} s_k^i(t) \quad (3.8)$$

where, $\parallel_{i=1}^{K_i}$ means concatenating $s_1^i(t) \parallel s_2^i(t) \parallel \dots \parallel s_{K_i}^i(t)$.

Our final reconstructed signal ($S^{R'}$) can be obtained by adding up the signals for each channel.

$$S^{R'}(t) = \sum_{i=1}^N s_i^R(t) \quad (3.9)$$

This is the regenerated sound and written as a sound file. The sound is normalized by multiplying 0.9.

$$S^R(t) = 0.9 S^{R'}(t) \quad (3.10)$$

The normalization of sound is necessary, because if a decoded sound signal is containing a value greater than 1, those values will be clipped by MATLAB.

Reconstructing the sounds from the original spike codes raises many issues which are discussed next.

3.3 The effect of Delay Vectors

The effect of the delay vectors, produced by GMF, is significant in this reconstruction work. These values are different for each channel. The delay vectors are D^i , where $i = 1, 2, \dots, N$, the times by which the signals are delayed in each channel.

Filterbanks are commonly used in sound coding-decoding system and they delay by both phase and group, where group delay is measured by the time delay of the amplitude envelopes of a sound signal and phase delay can be measured by the time delay of the phase in that sound signal ([71]). Phase delays are at the beginning of the signal, group delays are at the highest amplitude of the signal. The characteristics of delays are different for different filterbanks. The delay vectors are used in the reconstruction technique to produce the decoded signal from our biologically inspired spike code. This work has been extended to investigate the exact nature of the delay vectors, used in this biologically inspired spike based coding technique, as it has a considerable impact on reconstruction work. The original signal and the reconstructed signal differ significantly at the beginning of the signal and at the end of the signal. This examination of the delay vectors will help to illustrate that fact.

Figure 3.3 shows the nature and properties of delay vectors. A step function has been constructed and channel signals 1, 10, 15, 20 and 25 are chosen to compare the effect of the delay vectors in each signal.

The research claims the delay vectors are calculated to find the delay at the highest amplitude of a signal, as the equation (41) from page 69, [38] and as demonstrated in figure 3.3. This work has been done with other kind of sound signals like – triangular wave, guitar sound, and according to the linear system theory the output has been always the same.

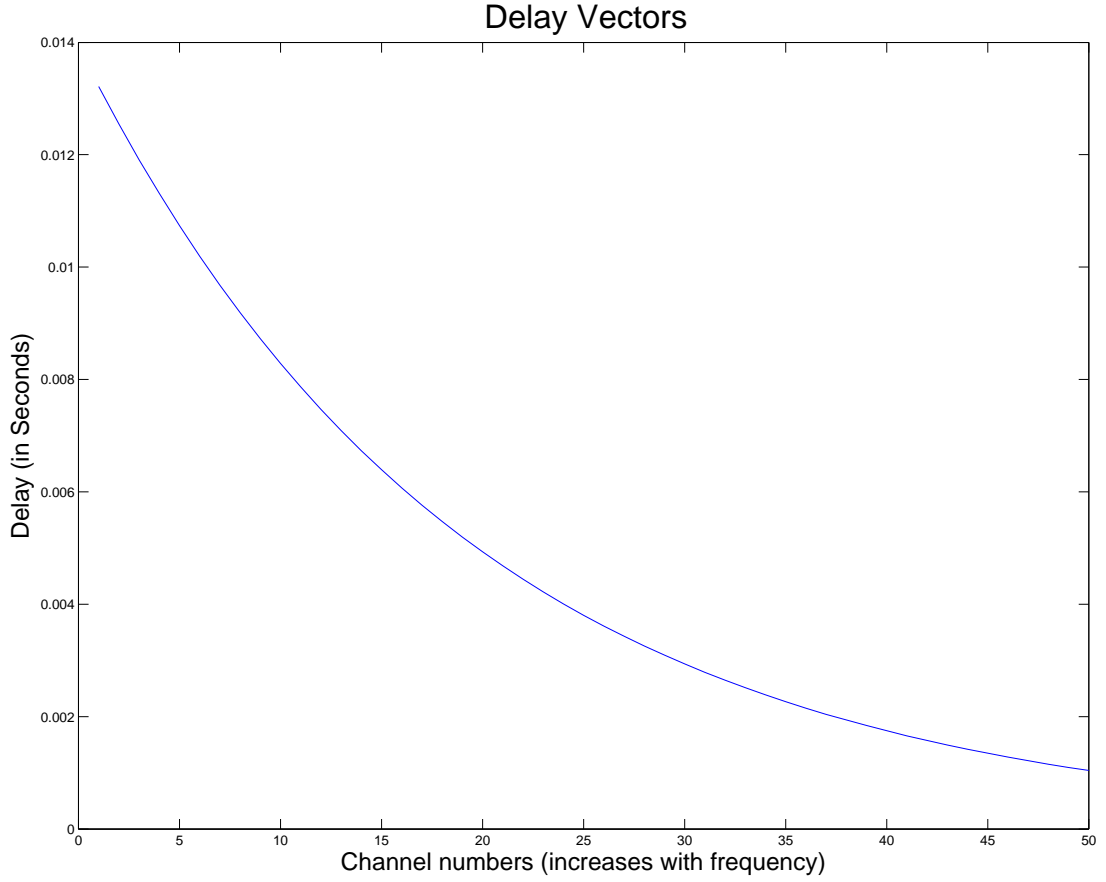


FIGURE 3.2: The plots of delay vectors by their values along with different filterbank channels, which increases with the increase of frequency. The channel frequency range is from 100 Hz to 10 kHz.

3.3.1 Investigating the nature of Delay Vectors

The nature of delay vectors has been investigated by plotting the decoded signal (decoded without delay compensation) along with the original signal. To reconstruct the signal, time values of delay vectors are subtracted from the original time value of occurrence of the spikes [see equation 3.1].

Recall that our sorted spike code is $Q(i) = \{t_n^i\}$, And if we refer to the equation 3.1, this relationship can be easily established that

$$Q(i) - Q^d(i) = D^i \quad \text{for } i = 1, 2, \dots, N \quad (3.11)$$

A step function has been used here to test the effect of the delay vectors in reconstructed signal as the step function crosses zero only once and the time of occurrence of spike is

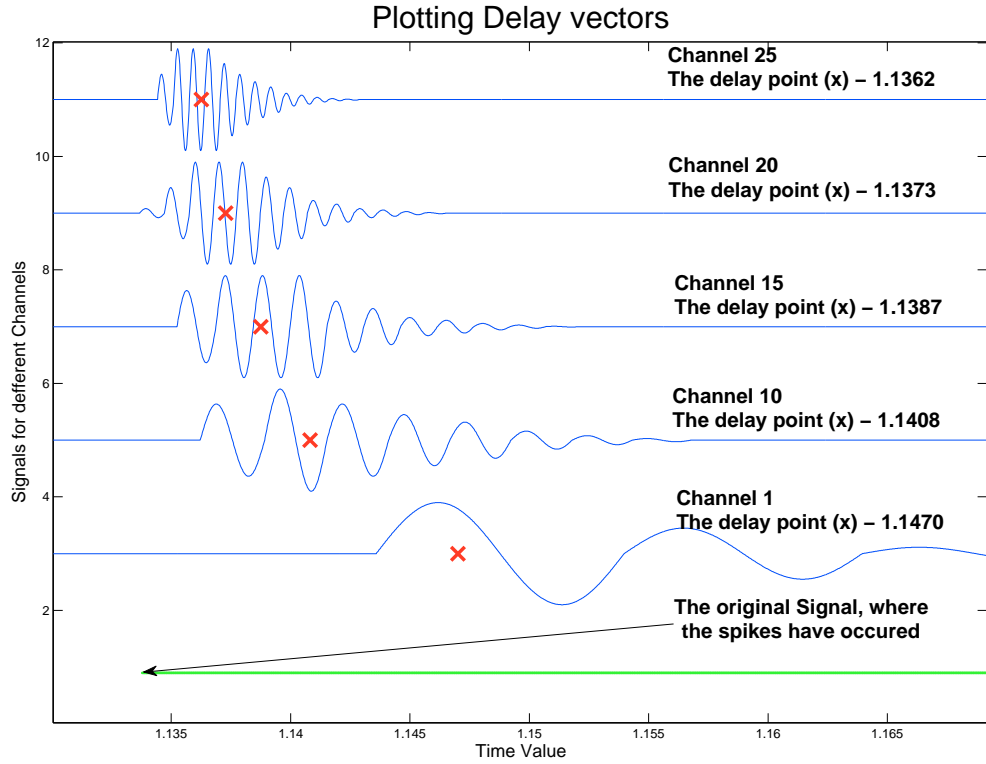


FIGURE 3.3: **The delay vector plotting for reconstructed signal (without delay compensation):-** Here it can be seen that the characteristic of the delays occur at the highest peak of amplitude of the reconstructed signal, roughly speaking. The step function steps from 0 to 1 at time 1.1338 sec. The red crosses(x) [delay point] represent the time value of the delay vector for each channel (see equation 3.11). The delay point is calculated by adding the time value of the zero crossing of the step signal and the delay vector in that particular channel. The green line at the bottom is the original step signal with the value as 1. It can be clearly seen that the ‘red cross’ [delay point] appears close to the highest peak of the amplitude of the each channel signal.

known. The delay of the reconstructed step signal for each channel can be clearly seen and be compared with the original step signal in a single graph. See figure 3.3.

In the figure 3.3, it can be seen that the GMF delays more for the lower frequency than the higher frequency. It decreases with the increase of channel center frequency. Delay vectors are inversely related with the channel numbers, which increases with frequency. So, $D^i \propto \frac{1}{n}$ and $n \propto i$, where, n is the channel numbers [1, 2, ... N] and i is the frequencies. So, $D^i \propto \frac{1}{f_c^i}$. So, $f_c^1 < f_c^2 < f_c^3 < \dots < f_c^i$ and $D^1 > D^2 > D^3 > \dots > D^i$, where f_c^i is cochlea center frequency and D^i is delay vector.

The effect of this delay-compensation is that the reconstructed signal sounds more similar to the original signal. However the reconstructed signal actually starts a bit earlier than the original signal [see figure 3.4]. This happens because of the characteristics of the

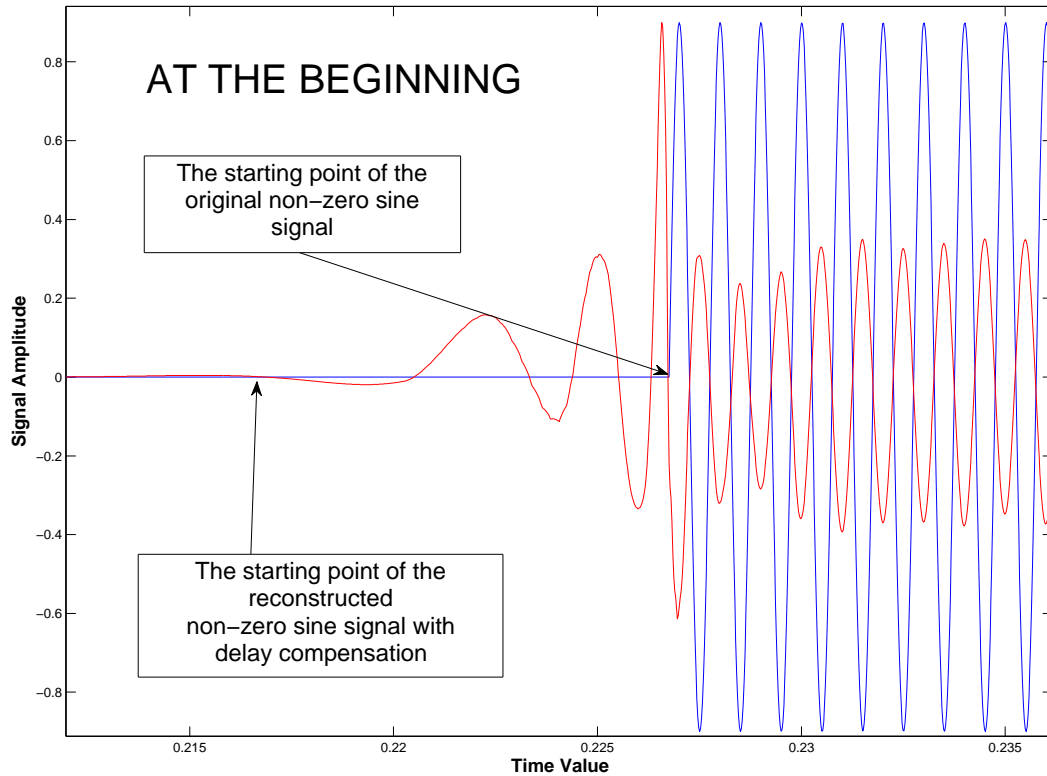


FIGURE 3.4: Comparison between reconstructed signal and the original signal (at the beginning): The reconstructed non-zero signal starts at 0.217 sec, whereas the non-zero original sine wave starts at 0.2267 sec. There is a 0.0097 sec error at the beginning, due to the nature of the delay vector and the reason discussed above. Significantly, it can be seen that before 0.2267 sec, the reconstructed signal has quite a smooth signal curve. This is because that low frequency signal has bigger delay than high frequencies. So, in the delay compensation work, the lower frequency ones come forward than the higher ones.

delay vectors. They are calculated to find the delay at the highest amplitude of the signal in a channel. Now if the highest amplitude point in the whole channel signal occurs at the beginning, delay-compensation method finds the starting of the reconstructed signal almost correctly with respect to the original signal. But if that highest amplitude point remains towards the end of the channel signal, delay-compensation method actually shifts the signal a little bit earlier than what it should be if the highest amplitude point was at the beginning. This causes the reconstructed signal to start a little bit earlier than the original signal. It happens because the delays are not calculated on the basis of the beginning of a signal, rather it is to produce the delay for the highest peak of amplitude in each channel. The delays decrease as the frequency of channels increase.

Figure 3.4 illustrates the facts discussed above. The graph takes a closer look of both original and reconstructed sound signal of a 1 kHz Sine wave at the beginning. So, in

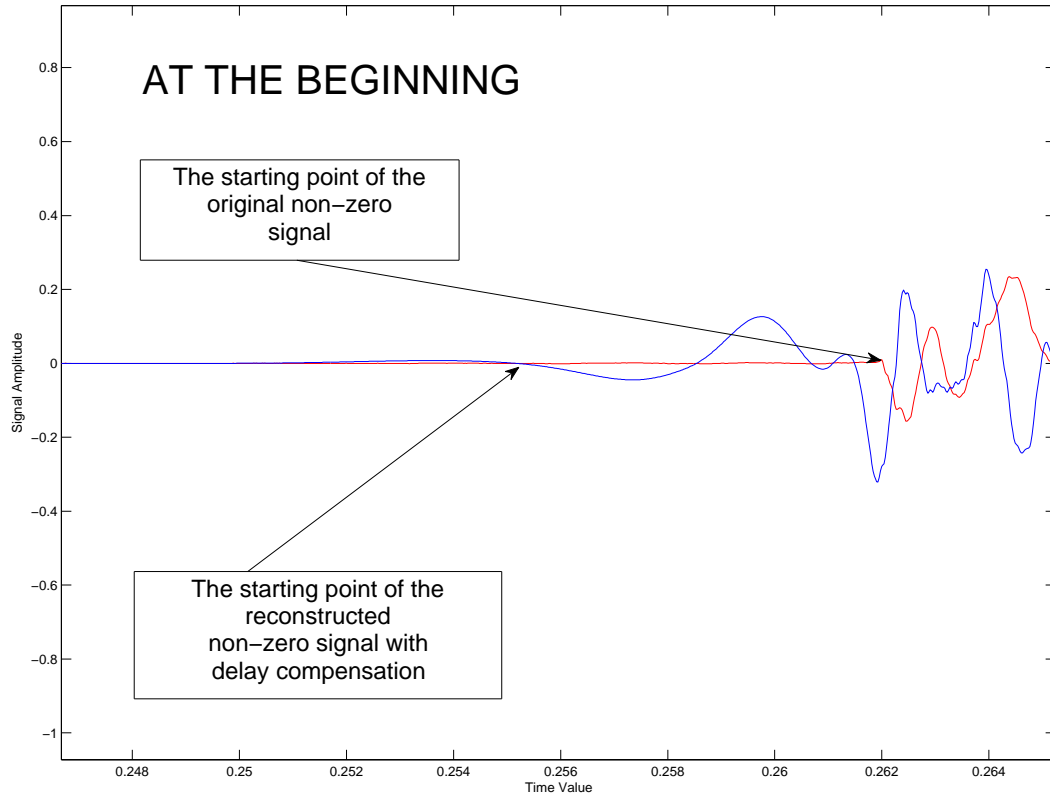


FIGURE 3.5: Comparison between reconstructed signal and the original signal (at the beginning): This time a natural sound like sound of cracking woods has been considered for comparison. The reconstructed non-zero signal starts at 0.2507 sec, whereas the non-zero original sound signal starts at 0.261 sec. There is a 0.0103 sec error at the beginning, due to the nature of the delay vector and the reason discussed above. Significantly, it can be seen that before 0.261 sec, the reconstructed signal has quite a smooth signal curve. This is because that low frequency signal has bigger delay than high frequencies. So, in the delay compensation work, the lower frequency ones come forward than the higher ones.

the delay compensation work, the lower frequency ones come forward more than the higher ones.

There is a bit of non-similarity, at the end as well, as the figure 3.6 illustrates. But that has nothing to do with the delay vector, rather the way our BM like GMF produces the bandpassed signal for different channels. The spikes occur in different sensitivity level for different threshold value on the band-passed signal for each channel. Now take an example of the same step signal used before (figure 3.3). The value of the step function changes from zero to one in a moment. (Precisely speaking the value of the step function rises to one from zero in less than $\frac{1}{f_s}$ sec, where f_s is the sampling rate (44100 samples/sec). So, indeed the step function rises in 0.0000227 sec. But the difference between the time values of the first and last spike in the spike code is 0.040 sec.) When we listen to the signal, it makes a quick ‘click’ sound. The duration of that

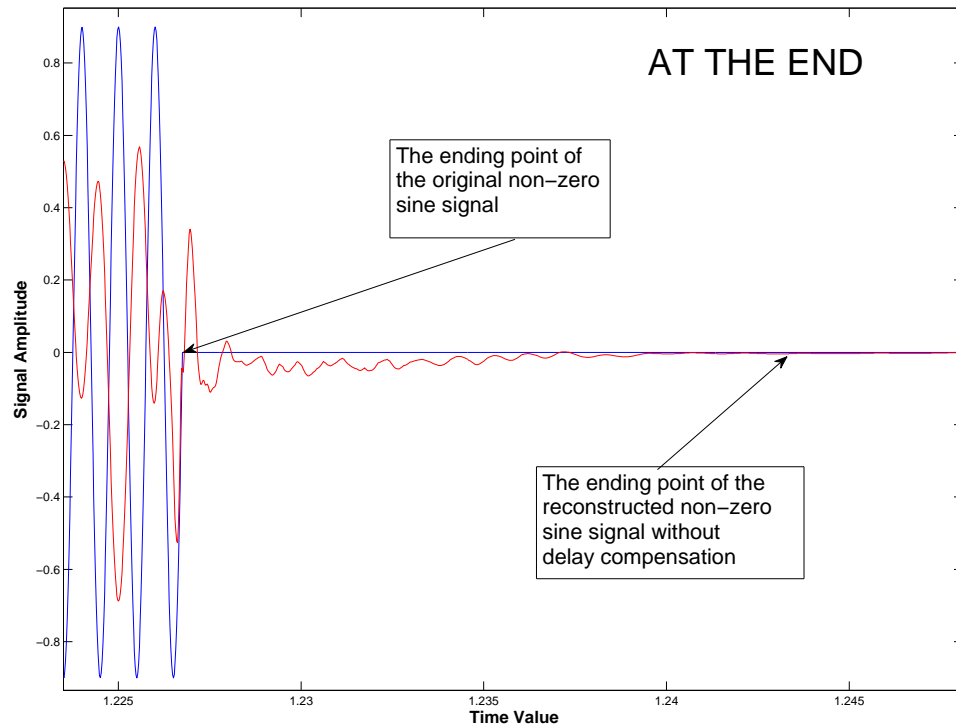


FIGURE 3.6: **Comparison between reconstructed signal and the original signal (at the end):** The non-similarities are present at the end as well. But this is not because of the Delay Compensation work. it is because the filter ‘rings’ a little.

‘click’ sound appears longer in our ear to produce spike in our auditory nerve than the microscopic time interval of amplitude rise. The biologically inspired spike based coding and decoding system supposed to produce spike in a way that our auditory system works. The reconstruction work is based on the produced spike code from the original sound signal. That’s why the reconstructed signal has slightly elongated non-zero signal compared to the original sound signal.

3.3.2 No Delay Compensation

Delay compensation method is appropriate to use in the reconstruction as it is produced by the GMF and it is a part of the spike code. If the decoded sound is not delay compensated, the starting point of the original and decoded signal will be the same as illustrated in figure 3.7.

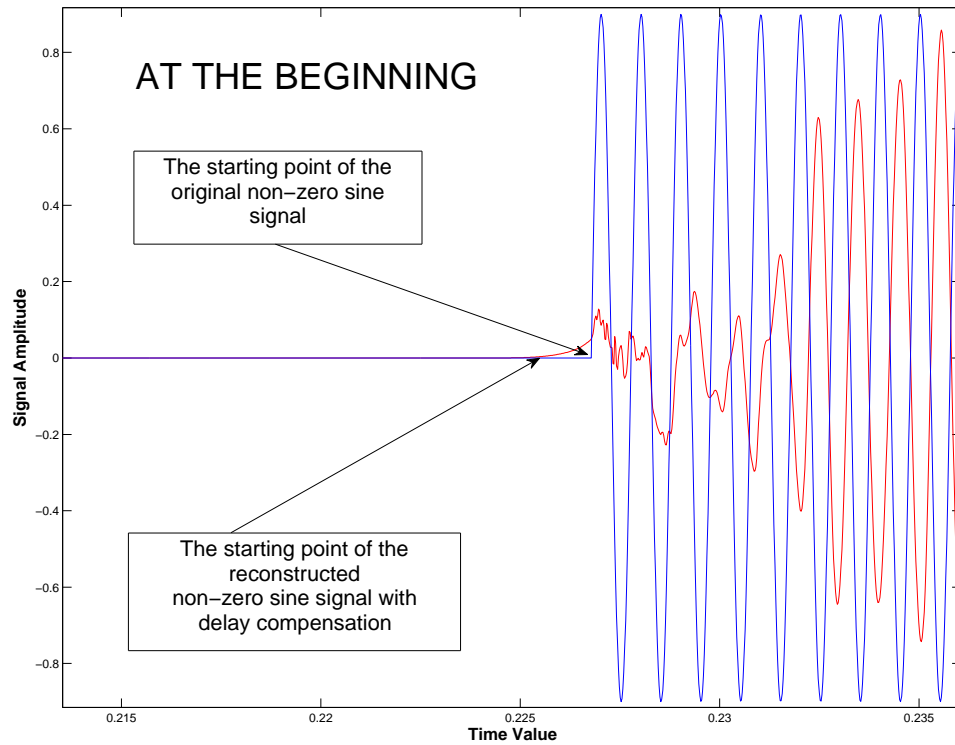


FIGURE 3.7: Comparison between reconstructed signal and the original signal at the beginning (without delay compensation):- Here, we can notice that the decoded sound has started along with the 1 kHz original sound, unlike figure 3.4

3.4 Filterbank Tuning for Low sampled Sound

The GMF requires right parameter values for high and low sampling rate sounds. In Timit Dataset ([72]), all the sounds are sampled at 16000 Hz. According to Nyquist frequency ([73]), the frequency has to be half of the sampling rate to avoid aliasing ([74]) as mentioned in equation 3.12.

$$f_k = 0.5 \times f_s \quad (3.12)$$

where f_s is the sampling rate of that sound.

Although the Nyquist frequency for 16000 samples/sec is 8000 kHz, it has been found that including sounds near 8 kHz resulted in poor quality resynthesis. We had to multiply the Nyquist frequency mentioned at equation 3.12 by 0.5 so that the GMF works properly for the low sampling rated sounds. So, originally in the MATLAB code, the highest cut-off frequency has been assigned as the minimum of the 10,000 Hz or $0.5 \times 0.5 \times f_s$ (mentioned at equation 3.13), where f_s is the sampling rate of that sound. In this way, the GMF produces the proper bandpassed signals for higher frequencies.

So, the adjusted cut-off frequency has been:

$$f_k = 0.5 \times 0.5 \times f_s \quad (3.13)$$

where f_s is the sampling rate of that sound.

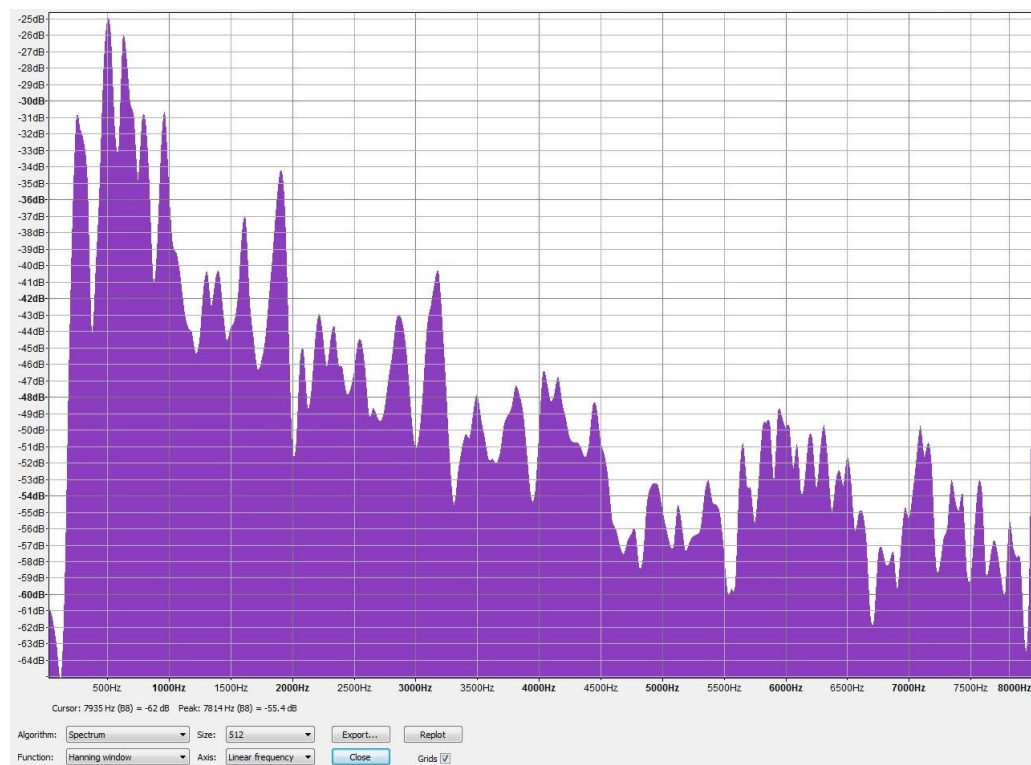


FIGURE 3.8: Spectrogram of the original sound: This spectrogram, taken from Audacity software for a female speech sound, shows that no the energy contents higher than 8 kHz, are present in the spectrogram. As the sounds are sampled at 16 kHz, the maximum frequency content in those sounds should be 8 kHz. The X-axis shows frequency ranges from 500 Hz to 8000 Hz. Y-axis shows sound intensity from -64 dB to -25 dB.

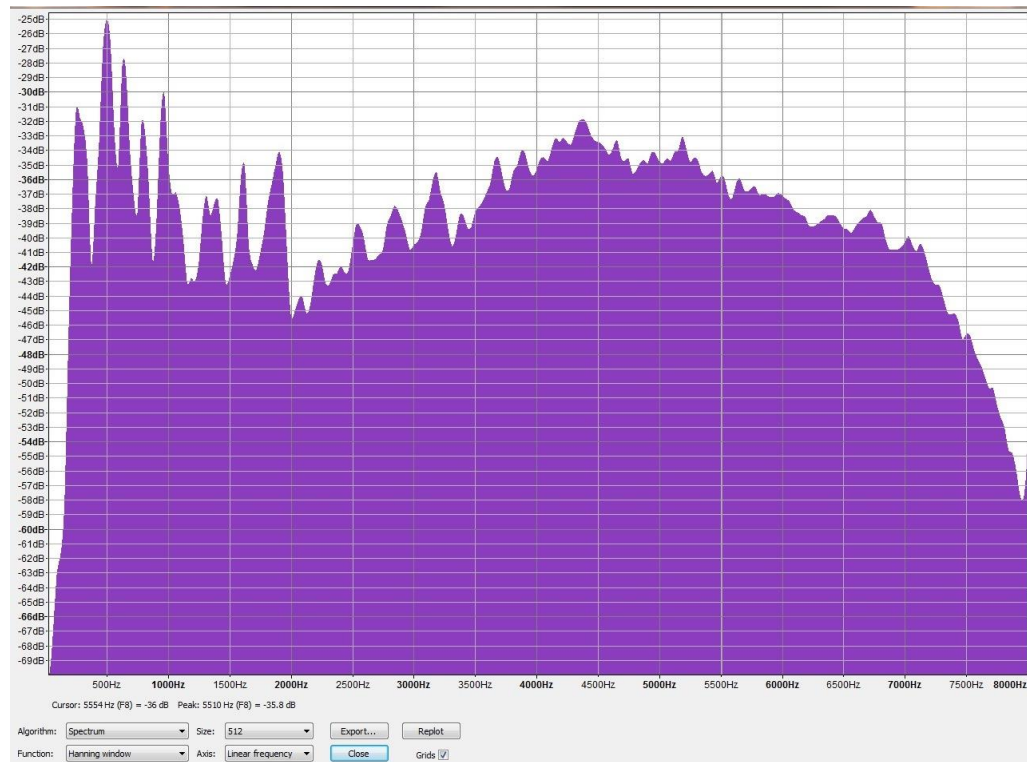


FIGURE 3.9: Spectrogram of the decoded sound of figure 3.8 according to Nyquist frequency in equation 3.12: This spectrogram from Audacity software shows that the energy contents are overly-distributed at the higher frequencies. The X-axis shows frequency ranges from 500 Hz to 8000 Hz. Y-axis shows sound intensity from -69 dB to -25 dB. That’s why, the decoded signal has extra high frequency noises present in them. This shows that in this case GMF is not properly tuned for low sample rates.

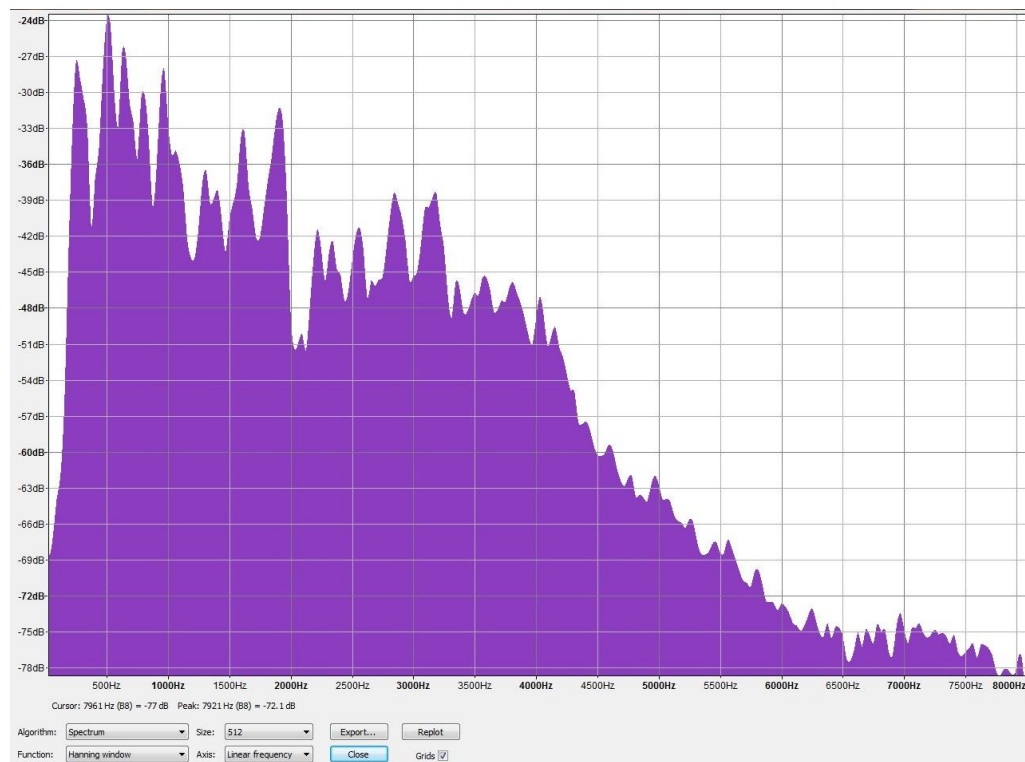


FIGURE 3.10: **Spectrogram of the decoded sound of figure 3.8 according to Adjusted frequency in equation 3.13:-** This spectrogram from Audacity software shows that the energy contents are below 8 kHz and the contents are better distributed at the higher frequencies. The X-axis shows frequency ranges from 500 Hz to 8000 Hz. Y-axis shows sound intensity from -78 dB to -24 dB. This sounds much more similar to the original sound.

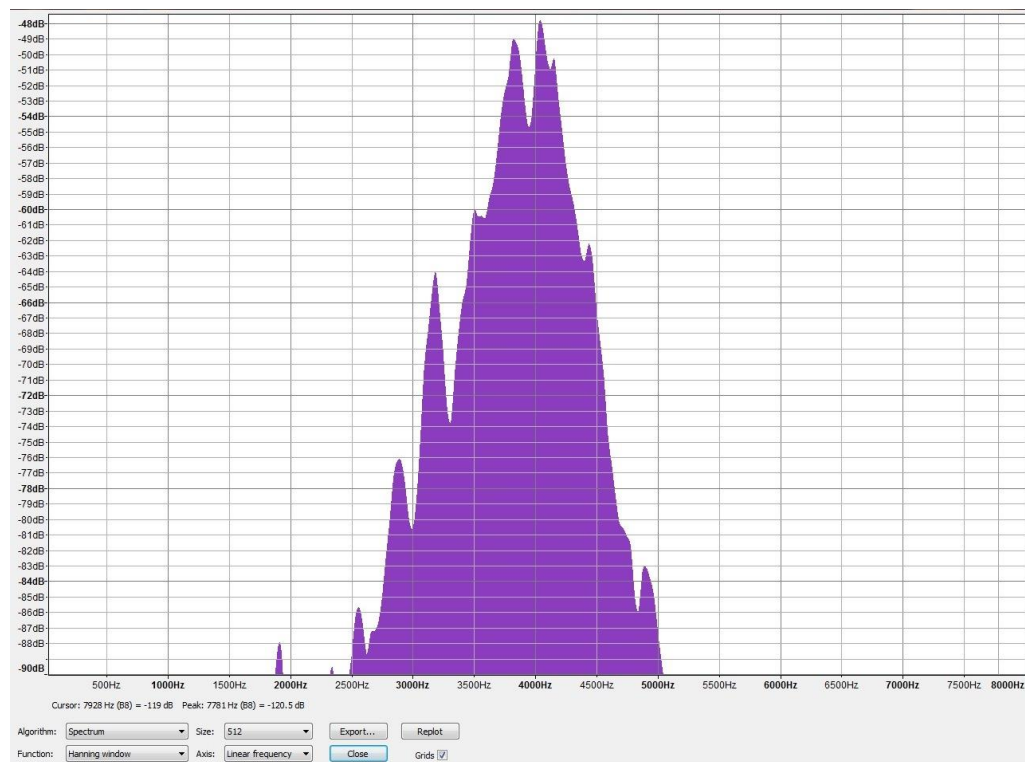


FIGURE 3.11: **Spectrogram of the bandpassed signal at the highest frequency according to the adjusted frequency in equation 3.13:** This spectrogram from Audacity software shows that the energy contents are at its peak at 4kHz. However, the energy continues up to 5 kHz. The X-axis shows frequency ranges from 500 Hz to 8000 Hz. Y-axis shows sound intensity from -90 dB to -48 dB. This causes to generate the decoded sound without any noise.

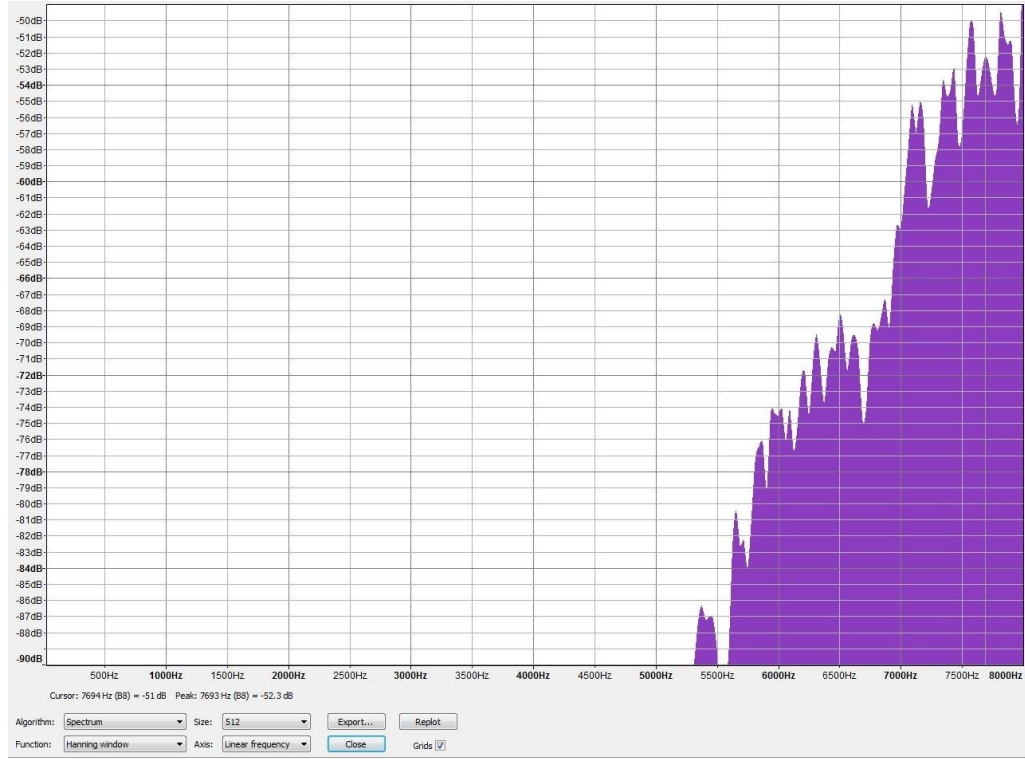


FIGURE 3.12: **Spectrogram of the bandpassed signal at the highest channel (frequency) according to Nyquist frequency in equation 3.12:** This spectrogram from Audacity software shows that the energy contents are at its peak at 8 kHz. However, the original sound does not have any energy contents higher than 8 kHz. This causes to generate extra energy contents in the decoded signal shown in figure 3.9, which in fact generates extra noises in the decoded signal as mentioned in figure 3.9.

The real issue is that by including frequencies, greater than $\frac{NyquistFrequency}{2}$, we end up generating too many high frequency contents. And by following equation 3.13, we can generate reconstructed sounds without any noises.

3.5 Number of Channels and Sensitivity Labels

The number of channels and sensitivity levels used in this biologically inspired spike coding technique is very significant to produce a noise free and well reconstructed sound. The advantage of using fewer channels and sensitivity levels is that it consumes less memory and disk space. To implement this spike based coding-decoding technique in electrical device for high number of channel and sensitivity level will be expensive as well. The channels are used to represent different frequency levels at which the original sound is bandpassed. The GMF used in ([3]) and ([4]) initially produces a range of frequency signal from 100 Hz to 10 kHz [so the frequency range becomes (10000 Hz-100 Hz) = 9900Hz], however it has been modified so that the maximum frequency of GMF becomes $\min(\frac{f_s}{4}, 10kHz)$ (according to equation 3.13 where f_s is the sampling rate). So, $f_c^1 = 100\text{Hz}$ and $f_c^N = \min(\frac{f_s}{4}, 10kHz)$, where N is the total number of channels. So, by using fewer channels the frequency-gaps become bigger, making the spike generation technique more lossy.

This is also true for number of sensitivity level used here. The sensitivity levels are used to represent the amplitude of previous quarter of the signal at the spike occurring time. In this spike coding technique the highest threshold level Θ_ξ is 0.03620 and the min threshold level Θ_1 is 0.0002 to consider a spike's sensitivity level as mentioned in section 2.3.2 at equation 2.3. So, from the equation 3.4, J_k^i becomes 0.0055 for $\hat{j} = 1$ and 1 for $\hat{j} = \xi$. Using fewer sensitivity levels suggests that this technique becomes more lossy as the range of information about the amplitude of original sound reduces. The appropriate sensitivity level of a spike is very important in this spike based coding-decoding system as the multiplier of the reconstructed sine wave is calculated on the basis of that sensitivity level.

Testing has been carried out in chapter 6 by increasing and reducing the number of channels and sensitivity levels in spike generation technique. The decoded signals from those different numbers of channels and sensitivity levels have been compared with the original sound and from all those comparisons, we can find out which number of channels and sensitivity levels are the best for coding AN spikes.

3.6 Maximum Spiking Rates (MSR)

Another factor in the reconstruction work is to introduce the MSR of human auditory nerve. One auditory nerve fiber can fire maximum 200 spikes in a second, that is a neuron cannot fire another spike for at least 5 milliseconds after firing a spike ([75]). In this spike based model, for higher frequency channels, spikes are generated at a much higher rate than 200 spikes/second. Initially we used all of the spikes in high frequency channel to reconstruct the sound. As this reconstruction work claims to be biologically inspired spike based coding technique, the idea of the MSR should be reflected in the reconstruction work by considering only 1 spike in 5 millisecond.

Also, by applying this idea, there can be fewer spikes generated for a sound signal reducing the size of the coded sound. The algorithm has been visually described in figure 3.13 and is mathematically described below.

In the reconstruction work, the center frequency of each channel signal f_c^i has been checked. If $f_c^i > 200Hz$, the sorting process for the spikes $\{Q(i) = \{t_n^i\} : i = 1, 2, \dots, 50 \text{ \& } j = 1, 2, \dots, 16\}$ is done. The sorted spike time values are represented as \hat{t}_k^i and the sorting technique is

$$\hat{t}_k^i = \hat{t}_{k-1}^i + \frac{1}{SPR_{MAX}} + \epsilon \quad (3.14)$$

where, $\hat{t}_k^i \in t_k^i$ & $\hat{k} = 2, 3, \dots, \hat{K}_i$ and \hat{K}_i is the number of sorted spikes in the i th channel. For $\hat{k} = 1$, $\hat{k} = k = 1$. And $\hat{t}_k^i \leq \hat{t}_{K_i}^i$. Here, SPR_{MAX} is the MSR, so $SPR_{MAX} = 200 Hz$.

Basically, this sorting process takes the very first spike time and then considers the next spike which appears only after 5 milliseconds time. So, the new sorted spike times are the subset of the original spike times i.e $\hat{t}_k^i \subset t_k^i$. So, the new recorded spikes t_k^i consist of the sorted spike times \hat{t}_k^i . So, $\{t_k^i\} = (\hat{t}_k^i, \hat{j}_k^i)$. By implementing this technique, the sorted spikes t_k^i becomes more similar to the real spikes found in the human auditory nerve.

Next, multiple sine waves are generated between those sorted spikes t_k^i in that channel center frequency f_c^i . Those sine waves can be represented by $\bar{s}_k^i(t)$, which is

$$\bar{s}_k^i(t) = \sin(2\pi T \frac{1}{\hat{t}_k^i - \hat{t}_{k-1}^i} (\bar{f}_c)^i_k) \quad (3.15)$$

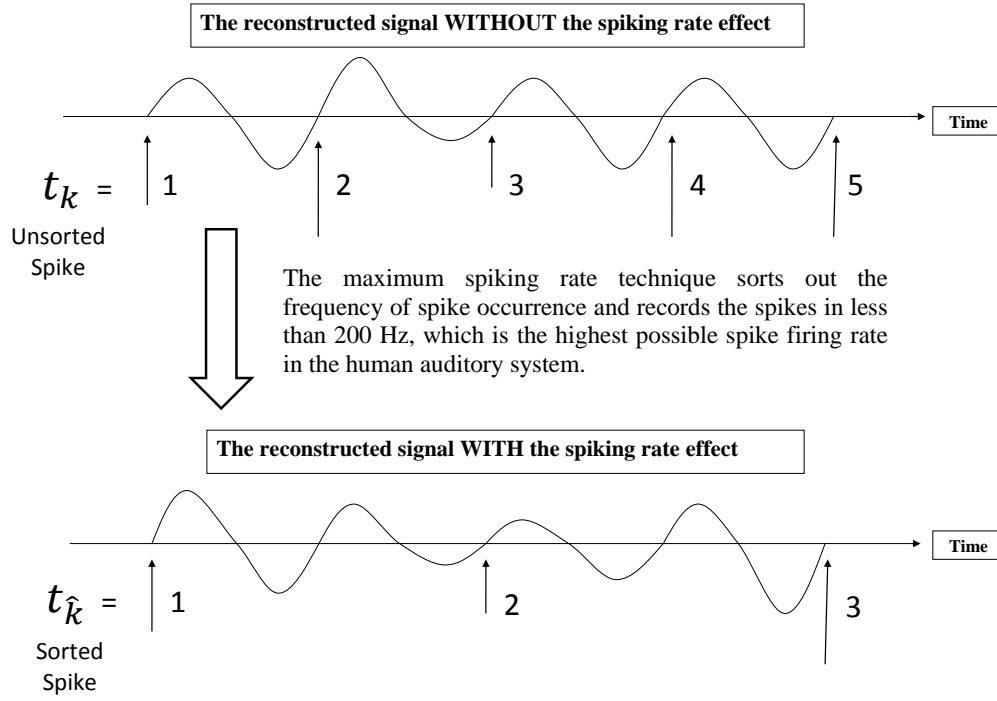


FIGURE 3.13: **Maximum Spiking Rates:** This technique sorts out the frequency of occurrences of spike rate in a particular channel. t_k are the unsorted spikes and $\hat{t}_{\hat{k}}$ are sorted spikes, where $k = 1, 2, \dots, 5$ and $\hat{k} = 1, 2, 3$. In this figure it can be seen that the spikes are recorded only after a certain time gap. That time gap has been calculated as 200 Hz, the maximum possible spike firing rate of human auditory nerve. It should be noticed that the sine waves are generated in the center frequency f_c^i in i th channel. But they are multiplied by the ramping multiplier \bar{J}_{k-1}^i to \bar{J}_k^i (see equation 3.4). This technique is very lossy.

where, $(\bar{f}_c)^i_k$ is the adjusted center frequency for this spike time interval which has been introduced only for this technique.

In this biologically inspired spike based coding system, $\hat{t}_k^i - \hat{t}_{k-1}^i \neq \frac{1}{f_c^i}$, rather $\hat{t}_k^i - \hat{t}_{k-1}^i \simeq \frac{1}{f_c^i}$. But, at spike times \hat{t}_k^i and \hat{t}_{k-1}^i , positive zero crossing is necessary. Let, the number of cycles in between \hat{t}_k^i and \hat{t}_{k-1}^i are Ω_k^i . The value of Ω_k^i is

$$\Omega_k^i = (\hat{t}_k^i - \hat{t}_{k-1}^i)(f_c^i) \quad (3.16)$$

where Ω_k^i is a Natural Number.

Now $\bar{\Omega}_k^i$ is chosen so that $\bar{\Omega}_k^i \leq \Omega_k^i$ and $\bar{\Omega}_k^i$ is the largest possible integer. So, $\bar{\Omega}_k^i \in \mathbb{N}$. For example, if Ω_k^i is 22.725, $\bar{\Omega}_k^i$ will be 22. So, the adjusted center frequency $(\bar{f}_c)^i_k$ becomes

$$(\bar{f}_c)^i_k = \frac{\bar{\Omega}_k^i}{\hat{t}_k^i - \hat{t}_{k-1}^i} \quad (3.17)$$

The sine wave $\bar{s}_k^i(t)$ is multiplied by the ramping multiplier \bar{J}_k^i . The value of \bar{J}_k^i can easily be derived from equation 3.20 which is $\{J_{\hat{k}-1}^i, (\frac{J_{\hat{k}}^i - J_{\hat{k}-1}^i}{R_{SM}}), 2(\frac{J_{\hat{k}}^i - J_{\hat{k}-1}^i}{R_{SM}}), \dots, J_{\hat{k}}^i\}$, where $\hat{k} = 2, 3, \dots, \hat{K}_i$. Here \hat{K}_i is the total number of sorted spikes and $\hat{K}_i < K_i$. For example, a 2 seconds long male speech and size of 178 KB sound generates 238108 spikes, but after this MSR technique, the number of spikes reduces to 17793 and the reconstructed sound is still easy to understand. Again for $\hat{k} = 1$, the value of \bar{J}_1^i is same like equation 3.21.

So, the new sine waves $\bar{s}_k^i(t)$ replaces the equation 3.5 in AN spike decoding algorithm, and the new equation becomes

$$s_k^i(t) = (\bar{J}_k^i)(\bar{s}_k^i(t)) \quad \text{for } \hat{k} = 1, 2, 3, \dots, \hat{K}_i \quad (3.18)$$

After that the equation 3.8 in AN spike decoding algorithm becomes

$$s_i^R(t) = \parallel_{k=1}^{\hat{K}_i} s_k^i(t) \quad (3.19)$$

where $s_1^i(t), s_2^i(t), s_3^i(t) \dots s_{\hat{K}_i}^i(t)$ has been concatenated (\hat{K}_i is the total number of spikes in that channel).

The effect of this MSR is very significant to this biologically inspired spike based technique. Figure 3.14 shows the spectrogram of the original sound file (which can also be played by clicking the caption in an electronic version of this thesis). However the decoded sound is not of good quality. The spectrogram in figure 3.15 shows that there is some extra energy which is concentrated at certain frequencies as the result of applying this MSR technique. Figure 3.16 explains why there is some extra energy concentrated at certain frequencies.

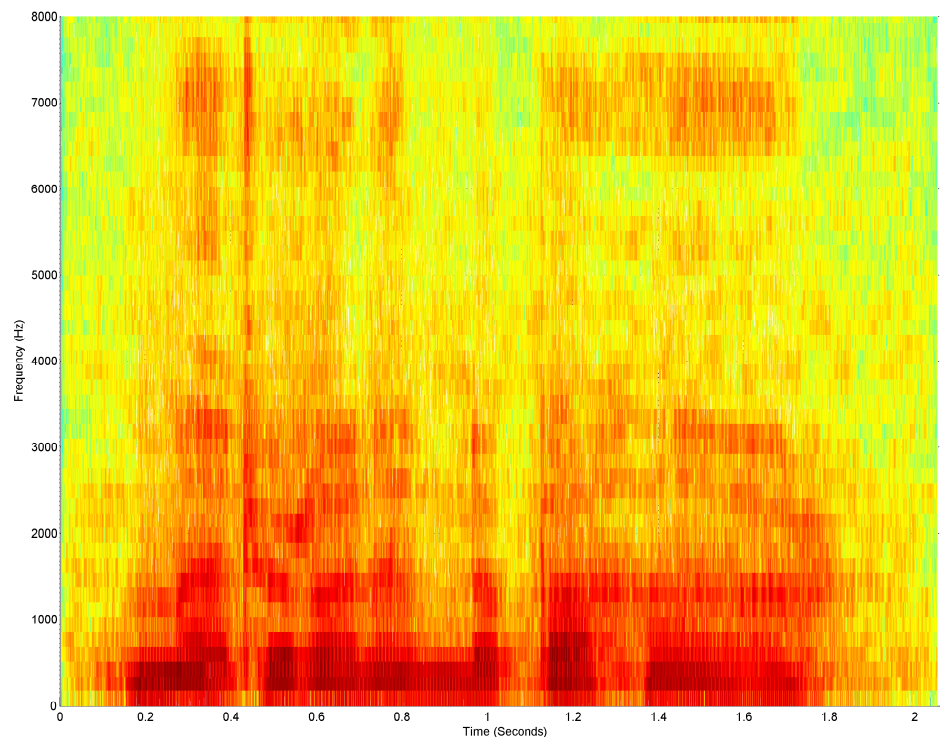


FIGURE 3.14: **The original sound signal:** The original sound. Listen to it here: ‘Sound Files Under Test/Speech/Test File (My Name)/testfile.wav’.

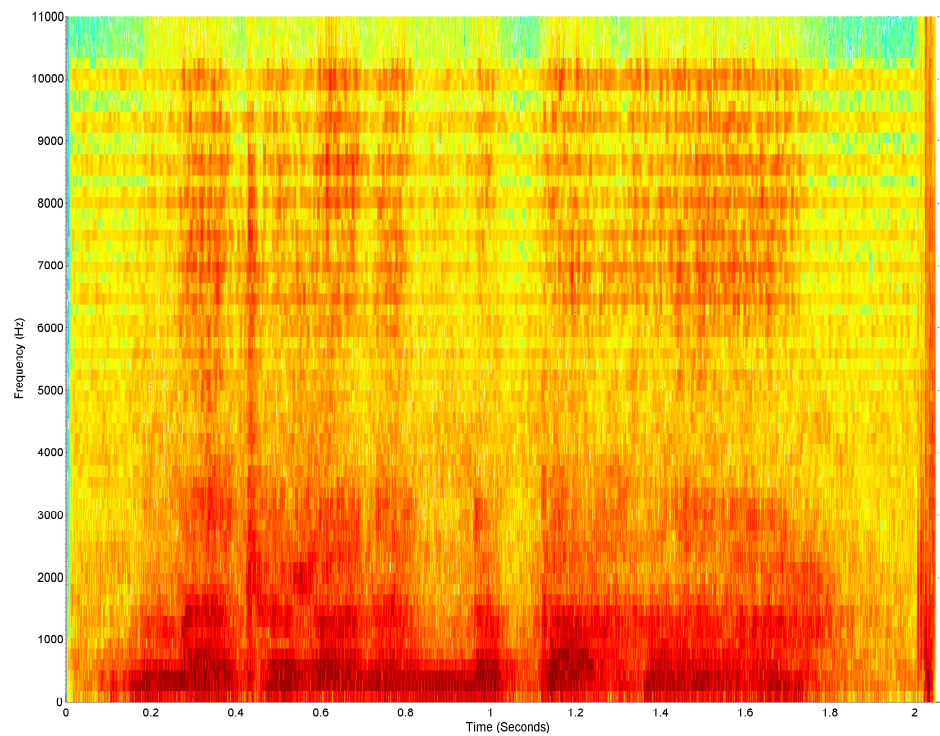


FIGURE 3.15: **The decoded sound signal:** The MSR has been applied to the original sound. Listen to it here: ‘Sound Files Under Test/Speech/Test File (My Name)/testfile_16_50_NEW_JTR0.wav’. Here we can see that the energies have been concentrated at certain frequencies. Figure 3.13 explains why this is the case.

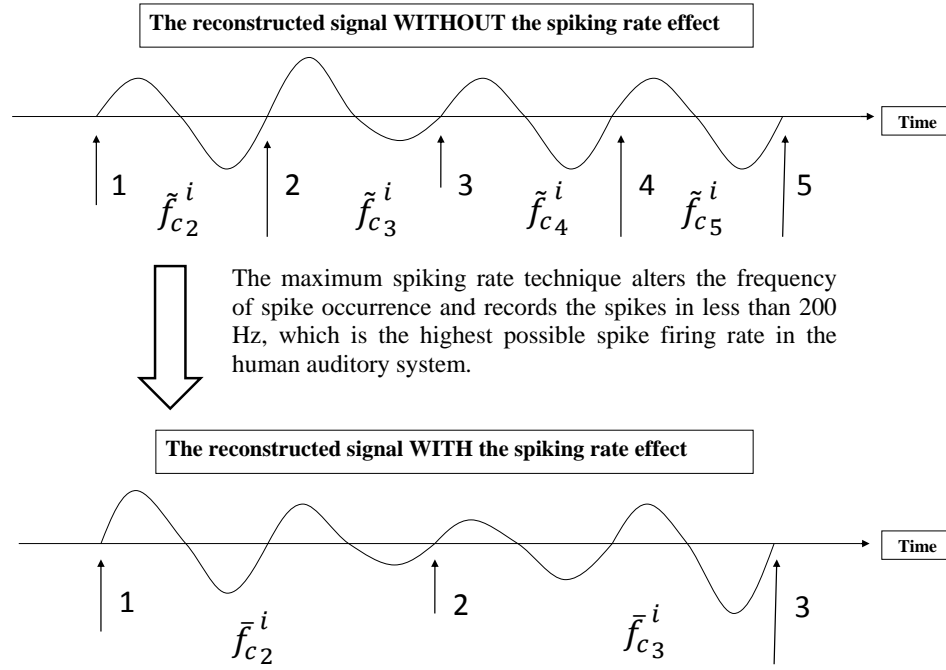


FIGURE 3.16: **The insight of the MSR:** $\tilde{f}_{c_k}^i$ is the frequencies for each sine wave generated in normal reconstruction. Here $\tilde{f}_{c_k}^i \sim f_c^i$ (f_c^i is the center frequency of GMF). But, $\bar{f}_{c_k}^i$ is the calculated frequencies, which are almost similar to f_c^i or $\bar{f}_{c_k}^i \simeq f_c^i$ ($k = 2, 3, \dots, 5$ & $\hat{k} = 2, 3$). $\bar{f}_{c_k}^i$ is much closer to f_c^i than $\tilde{f}_{c_k}^i$. This fact explains why there are some extra energy bands in certain frequencies. So, in the reconstructed sound there are some extra energy densities at the frequency level of each channel.

The original and final reconstructed sound can easily be compared by the figure 3.17 and figure 3.18. Also, in the figures, it can be seen that the amplitudes are not exactly same, though they are quite similar. In the spectrogram of the reconstructed signal, there is some extra energy bands have appeared. Again, figure 3.16 explains why there are some extra energies present in the decoded signal.

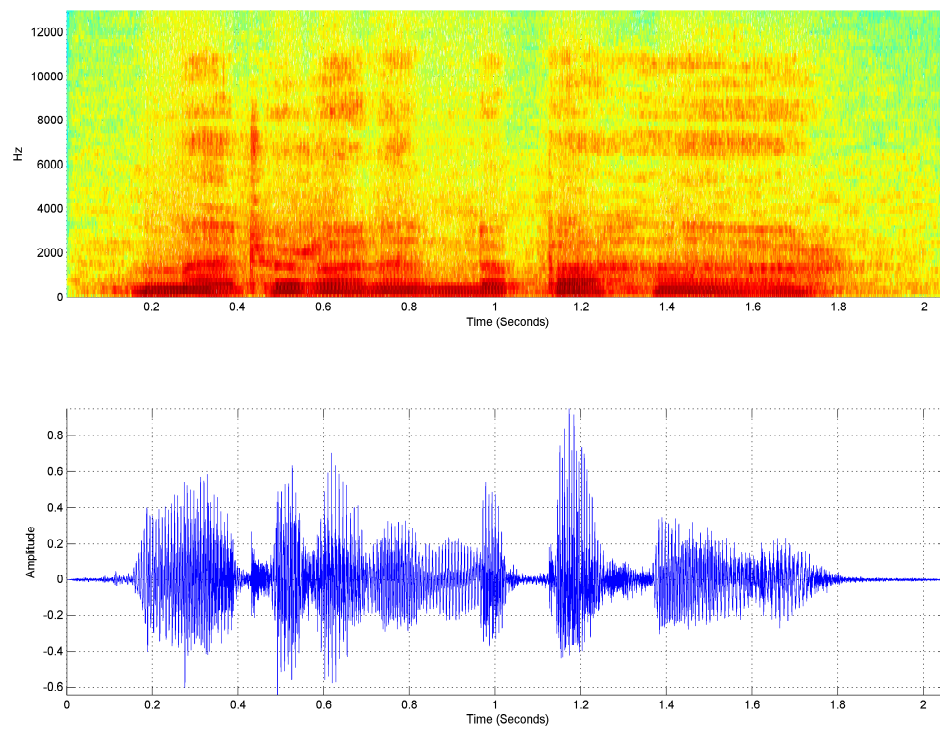


FIGURE 3.17: The original sound signal: The original sound is a Human Speech found at ‘Sound Files Under Test/Speech/Test File (My Name)/testfile.wav’. The top figure is the spectrogram and the bottom one is the signal amplitude of that sound.

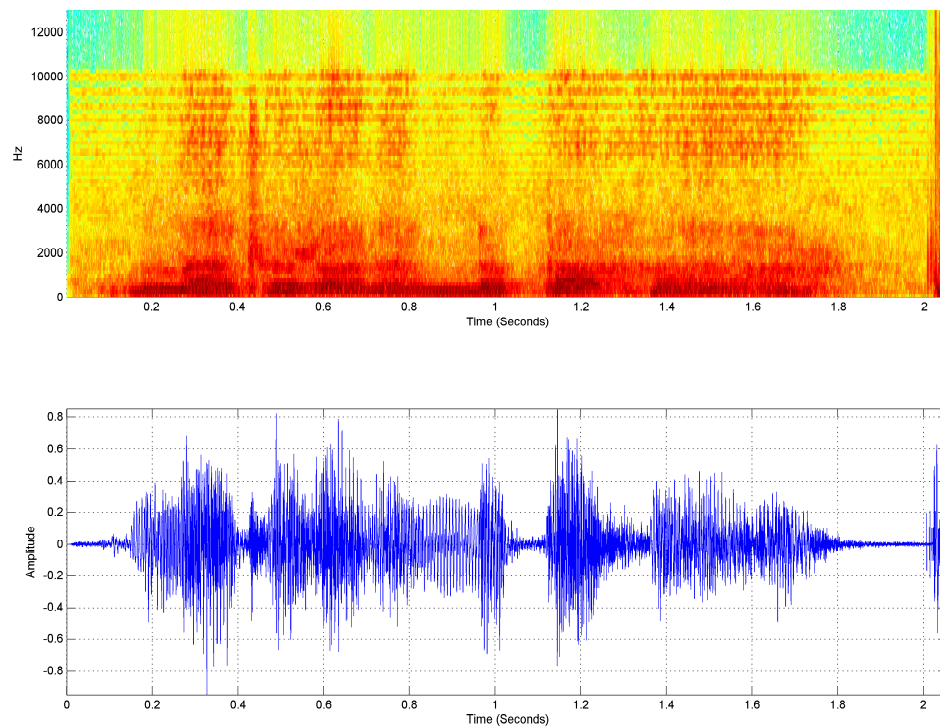


FIGURE 3.18: The reconstructed sound signal (With MSR technique): The reconstructed sound of that original sound found at ‘Sound Files Under Test/Speech/Test File (My Name)/testfile_16_50_NEW_JTR0.wav’. The sound is reconstructed with applying the MSR and Ramp Technique.

The MSR technique can be a big advantage to this spike based coding system. This proves that how efficient it can be if we use the biologically inspired technique. As, our human auditory nerve cannot fire spike at a higher rate than 200 per second, the loss in the sound signal due to this fact is not decisive. It reduces the number of spikes greatly and is very useful in AER system.

To rectify this concentration of energies at certain frequencies (as explained in figure 3.16), the occurrences of spikes have been jittered up to a certain amount. That is described in the next section.

3.6.1 Jittering the AN Spikes

MSR produces the some extra energies at the center frequencies of the GMF channels in the spectrogram of the reconstructed sound (can be found at ‘Sound Files Under Test/Speech/Test File (My Name)/testfile_16_50_NEW_JTR0.wav’ in the attached DVD) as shown in figure 3.19. Randomly jittering the frequencies between each spike time gap might solve this problem so it has been applied. Frequencies have been jittered to spread the extra energy distribution away from the relevant frequency band. As $\bar{f}_{c_k}^i$ is the frequency of a single sine wave in a channel signal, $\bar{f}_{c_k}^i$ has been randomly changed by σ , where σ varies from 1% to 10%.

Figure 3.20 & 3.21 show that the energies have been distributed a little bit away from the concentrated bit, but the sound quality deteriorates from the original sound quality.

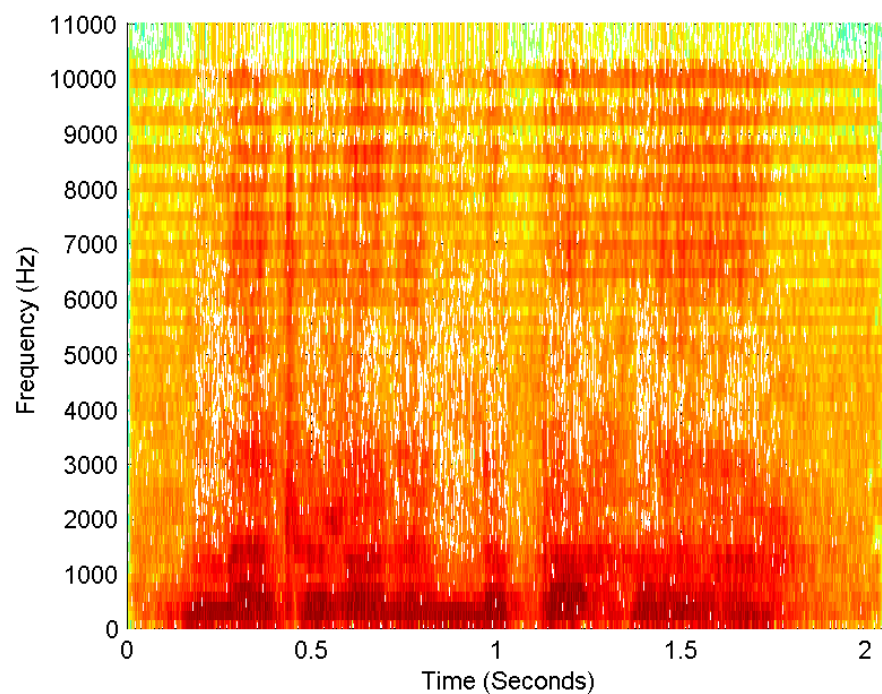


FIGURE 3.19: **Not Jittered at all** : This figure is similar with figure 3.15. The MSR has been applied to the original sound. Listen to it here: ‘Sound Files Under Test/Speech/Test File (My Name)/testfile.16.50_NEW_JTR0.wav’

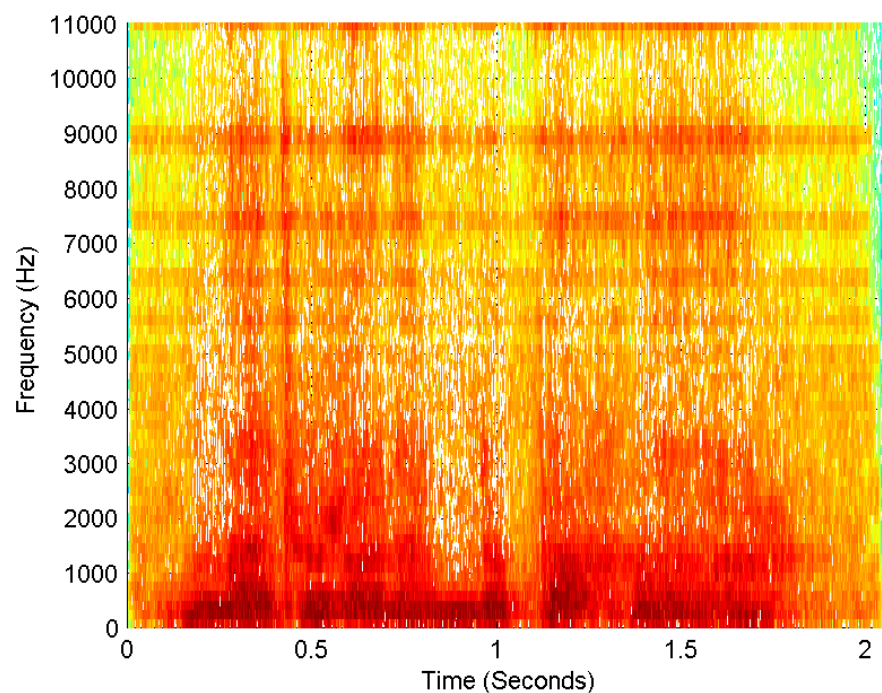


FIGURE 3.20: **Jittered by 1%**: The frequencies are randomly jittered by 1%. So, the spectrogram gets rid of some extra energy contents in compare to the figure 3.19. But does it sound better than before? Listen to it here: ‘Sound Files Under Test/Speech/Test File (My Name)/testfile.16.50_NEW_JTR2(1%).wav’

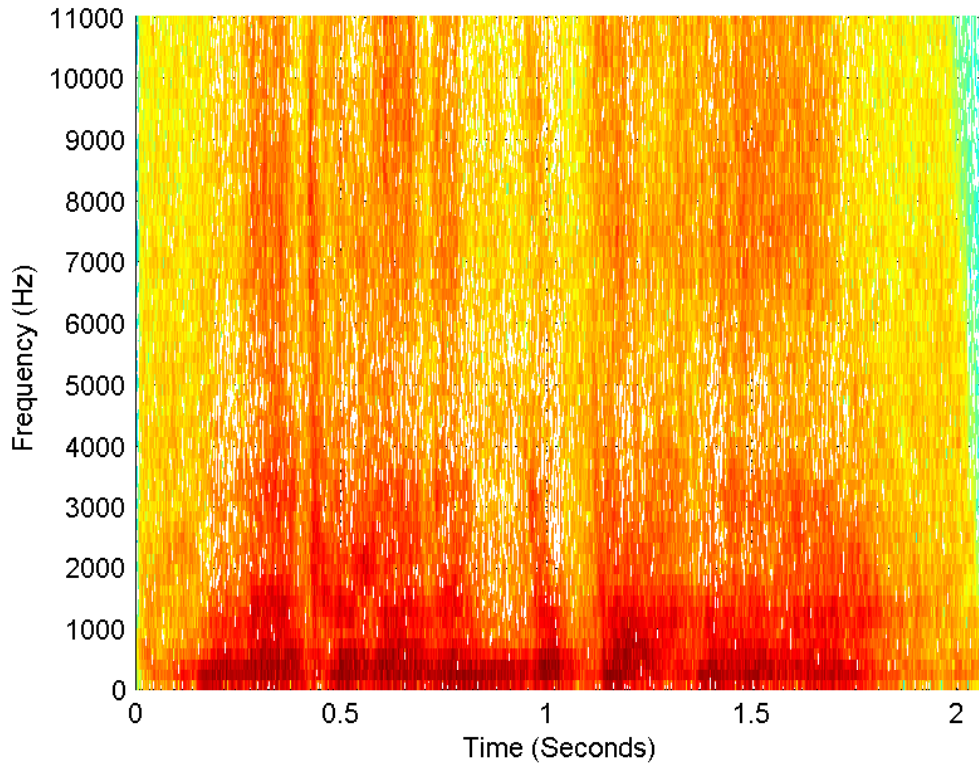


FIGURE 3.21: **Jittered by 10%:** The frequencies are jittered by 10% and the concentrated energy contents almost spreads everywhere in this spectrogram. But it does not help the sound to be better or similar to the original sound signal at all. If we take a closer look to this spectrogram and compare to the original sound's spectrogram, we will see that there is a pattern in the energy contents in the original sound. But jittering just randomly scatters the extra energy contents everywhere. We do want to get rid of the extra energy contents, but certainly not decreasing the sound quality. Listen to it here: 'Sound Files Under Test/Speech/Test File (My Name)/testfile_16_50_NEW_JTR2(10%).wav'

As the spikes have been jittered at 10%, according to the figure 3.21, the energies are widely spread all over and there is very little similarity with the spectrogram of original sound in figure 3.14.

Figures 3.20 & 3.21 and corresponding reconstructed sounds provide evidences that if we randomly jitter the frequencies, the energy contents do get distributed throughout the whole frequency range, but it reduces the quality of the sound. We cannot use this technique to eradicate the effect of MSR.

3.7 Further Issues in De-coding of AN Spikes

Further issues have raised in decoding of AN spikes. Those issues have been corrected as described below.

3.7.1 Ramp Technique

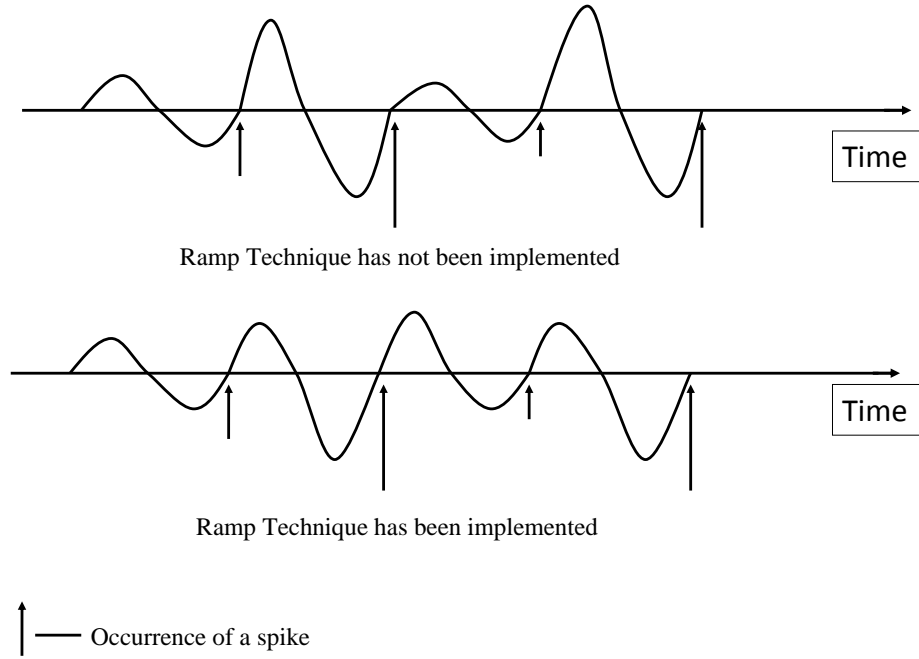


FIGURE 3.22: **Ramp Technique:** In the first case, the sensitivity levels of two consecutive spikes are quite different. So, there are some sharp changes at the zero crossings leading to extra harmonics. The second one, the ramp technique is applied. The discontinuities disappear, as instead of multiplying the whole sine wave by a number, a gradually ramping sequence (\bar{J}_k^i in equation 3.20) is multiplied by the sine wave. The length of these arrows representing spikes increases with the number of sensitivity levels in which the spike occurred.

The linear ramp technique has been implemented to the multiplier J_k^i (see equation 3.4) to multiply each sample of sine wave $\hat{s}_k^i(t)$. So each J_k^i becomes to a ramp multiplier \bar{J}_k^i , which is

$$\bar{J}_k^i = [J_{k-1}^i, (\frac{J_k^i - J_{k-1}^i}{R_{SM}}), 2(\frac{J_k^i - J_{k-1}^i}{R_{SM}}), \dots, J_k^i] \quad (3.20)$$

for $k = 2, 3, \dots, K_i$. For $k = 1$ equation 3.20 becomes

$$\bar{J}_1^i = [0, (\frac{J_1^i}{R_{SM}}), 2(\frac{J_1^i}{R_{SM}}), \dots, J_1^i] \quad (3.21)$$

And equation 3.5 becomes-

$$s_k^i(t) = (\bar{J}_k^i)[\hat{s}_k^i(t)] \quad \text{for } k = 1, 2, \dots, K_i \quad (3.22)$$

Previously, each sine wave \hat{s}_k^i was simply multiplied by the multiplier J_k^i . If $|J_{k+1}^i - J_k^i|$ is large, some harmonics will appear in between those two signals when concatenated together (see figure 3.22). Now each \hat{s}_k^i is multiplied by \bar{J}_k^i , which is the array of gradually increasing or decreasing numbers starting from J_{k-1}^i to J_k^i . So if the first signal is multiplied by a linearly ascending vector from J_{k-1}^i to J_k^i (if $J_{k-1}^i < J_k^i$) and the second one by a linearly descending vector from J_k^i to J_{k+1}^i (if $J_k^i > J_{k+1}^i$); the energy of the harmonics produced in this case will be much smaller.

Figure 3.23 shows the spectrogram where implementing this ramp technique compared to the original sound's spectrogram in figure 3.14. Figure 3.15 shows how the spectrogram will be with MSR but without ramp technique and figure 3.24 shows how the spectrogram will be without either MSR or ramp technique.

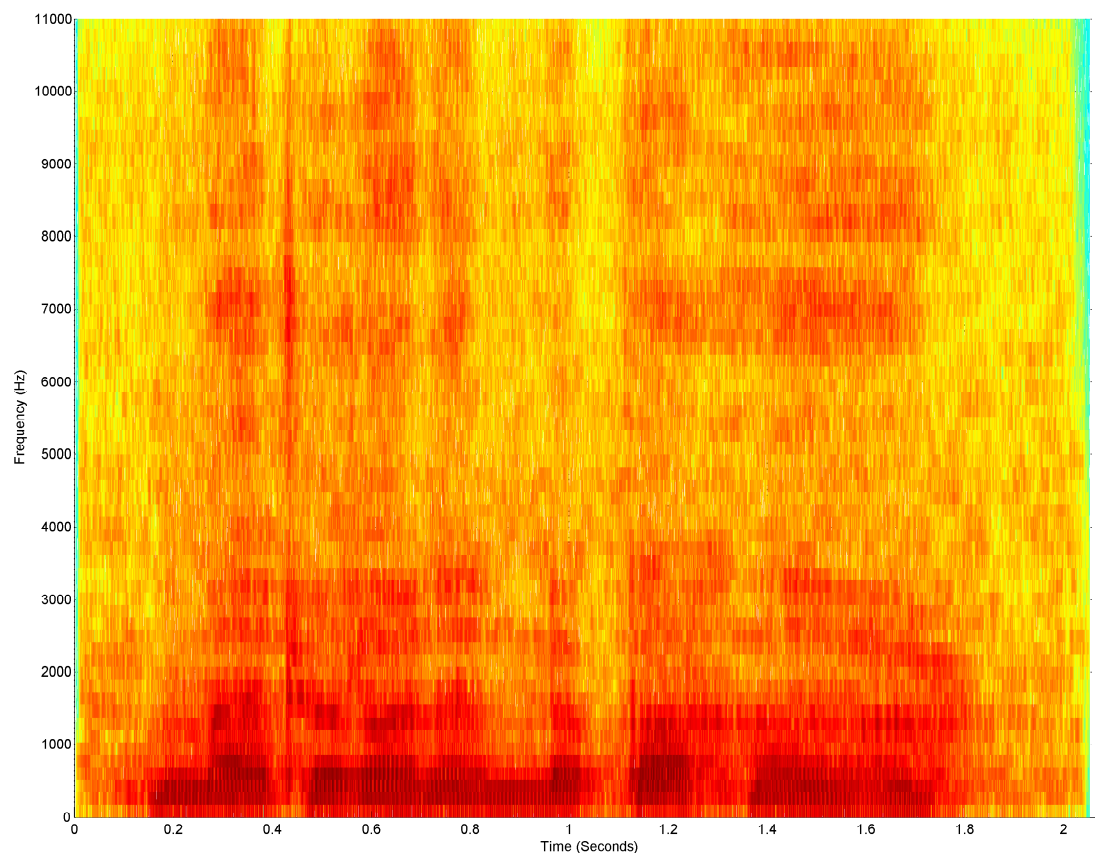


FIGURE 3.23: Spectrogram of the reconstructed sound signal (without MSR technique, with ramp technique) which is similar with the original sound in figure 3.14.

So, we can say by comparing the spectrograms in figure 3.23 and figure 3.24, that by implementing Ramp technique we cannot visually find any major differences either in the spectrogram or in sound quality, however it is sensible to apply it, because this does reduce artificial spectrogram components.

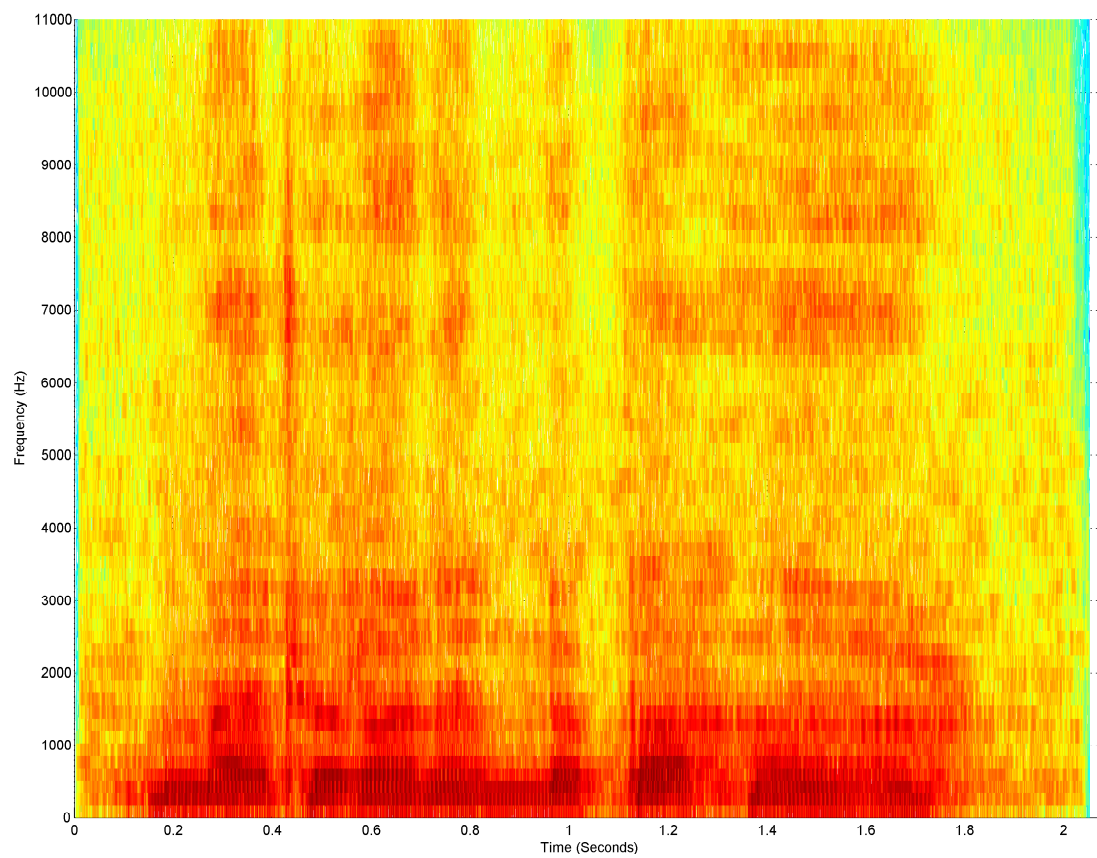


FIGURE 3.24: Spectrogram of the reconstructed sound signal (without MSR technique, without ramp technique). This time the spectrogram confirms that reconstructed sound is much similar with the original sound.

3.7.2 Smoothing Technique

This technique is applied to remove extra harmonics from the beginning and the end of a sound signal. In the coding technique, a minimum mean voltage level E_i has been considered in the previous quarter cycle of the bandpassed signal to occur a spike as mentioned in equation 2.3. Now if in the original signal there is any voltage which is less than the minimum voltage level, there is no spike at that time. In real world sound, often a large voltage appears next to a low voltage which may not be identified by a spike. So, the reconstructed sound signal from the spike code will not be able to produce that low voltage signal just before large voltage signal. Figure 3.25 describes this fact.

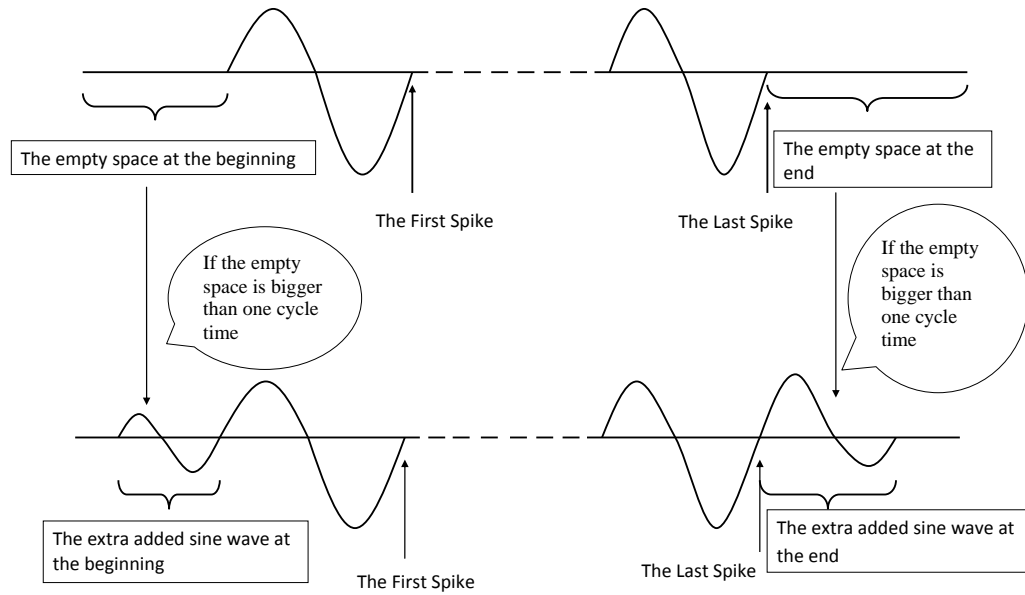


FIGURE 3.25: **Smoothing Technique:** Extra sine waves are added at the beginning and at the end to smooth the signal, which reduces extra harmonics energy at the beginning and end of the sound.

To minimize the sharpness of the reconstructed signal, some extra signals are added at the beginning and at end of each reconstructed channel signal $S_i^R(t)$ mentioned in equation 3.10. This work makes sure that the signal has a smooth start and finish. In the reconstruction work, the reconstructed channel signal often starts or finishes with a very sharp edge, especially when the first or last spike in that channel has occurred in a high sensitivity level. This produces unwanted harmonics at the start or end of the reconstructed sound signal. To solve this, another sine wave has been generated in the

center frequency of that channel and multiplied by an increasing vector like $\{0: J_1^i\}$ at the beginning of the channel signal and a decreasing vector like $\{J_{K_i}^i: 0\}$ (K_i is the total number of spikes in i th channel) has been multiplied at the end of that channel signal (see equation 3.20). The length of those sine waves is $\frac{1}{f_c^i}$.

Sometimes there is not enough space to add those ramping sine wave signals, if the spike occurs at the very beginning i.e. $\hat{t}_1^i < \frac{1}{f_c^i}$ or at the end of that channel signal i.e. $|T_{LN} * R_{SM} - \hat{t}_{K_i}^i| < \frac{1}{f_c^i}$. So in each case, this issue has been checked for each channel signal. Unfortunately if that happens, instead of concatenating that sine wave, only zeros of length $|\hat{t}_1^i * R_{SM} - 1|$ will be concatenated at the beginning. If there is not enough space available at the end, the sine wave with the frequency $\frac{1}{|T_{LN} * R_{SM} - \hat{t}_{K_i}^i|}$ will be concatenated at the end. The reason for adding the sine wave rather than adding zeros at the end is that the last spike in a channel often occurs at high sensitivity level. So, to remove the sharpness at the end, the sine wave has been concatenated rather than just zeros. In this case, the sound has been chopped off while it is still running, leading to less harmonics energy added at the end and even the original signal will have additional harmonics.

3.7.3 Spike Gaps

Sometimes, the original sound signal might have a long silence within it and that causes a long gap between the time values of occurrences of two consecutive spikes. Now, the methodology of the reconstruction work has been designed to have a single sine wave between two consecutive spikes. The frequency of the sine wave decreases as the time gap between occurrences of two spike increases. Now if the time gap is huge, there will be a single sine wave of very low frequency, producing inappropriate frequencies in the reconstructed signal. This issue has been solved as a special case in figure 3.26.

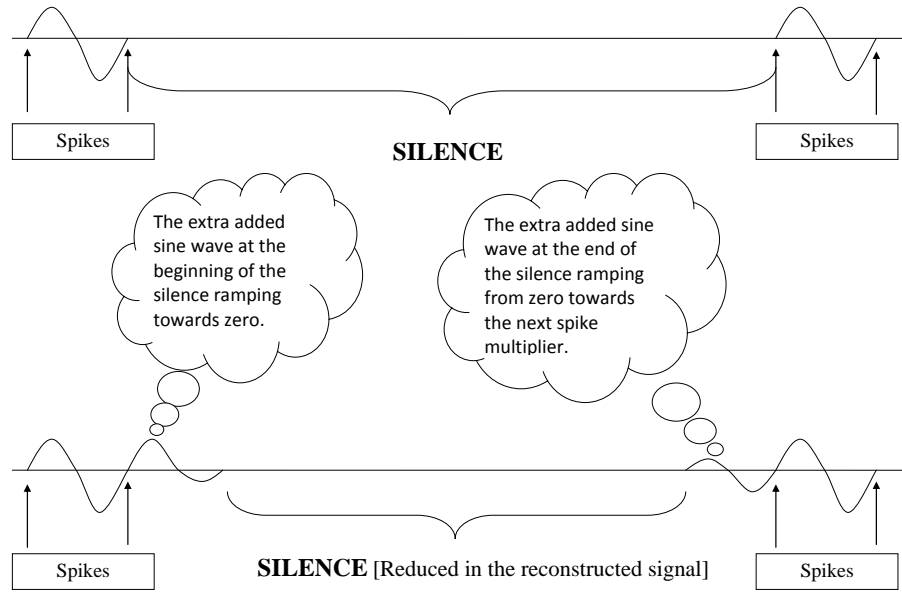


FIGURE 3.26: **Spike Gaps:** For a large time gap between the occurrences of two consecutive spikes, the silence in the reconstructed signal has been made. Two extra sine waves have been added in between the silence to reduce the possibility of unwanted harmonics occurring.

Let us consider that there is a large gap between the time value of occurrences of spike \hat{t}_k^i and \hat{t}_{k+1}^i in i th channel. So, the difference between the occurrences of those two spikes are $|\hat{t}_{k+1}^i - \hat{t}_k^i|$. Now if $|\hat{t}_{k+1}^i - \hat{t}_k^i| > \frac{2}{f_c^i}$, it has been considered that there is a large gap between two consecutive spikes. In this case, two sine waves of the center frequency f_c^i , will be concatenated just after \hat{t}_k^i and just before \hat{t}_{k+1}^i . Also the first sine wave will be multiplied by the decreasing vector $\{J_{k_1}^i : 0\}$ and the second sine wave will be multiplied by the increasing vector $\{0 : J_{k_2}^i\}$ (see equation 3.20). In between those two sine waves,

some zeros will be concatenated to represent that silence in the reconstructed signal. So, the length of the concatenated zeros i.e. silence is $|(\hat{t}_{k+1}^i - \hat{t}_k^i) - \frac{2}{f_c^i}|$.

3.7.4 A Single Spike in a Channel

In some cases, only one spike occurs in a certain channel signal. This time the reconstruction system does not work, as to generate a sine wave at least two spikes are necessary. The time value of the occurrence of that spike can be represented as \hat{t}_1^i . To solve this issue, two sine waves of the center frequency f_c^i of that channel, have been concatenated just before and after \hat{t}_1^i . The multiplier for the first sine wave is an increasing vector $\{0: J_1^i\}$ and the multiplier for the second one is a decreasing vector $\{J_1^i: 0\}$ (see equation 3.20). Figure 3.27 describes this.

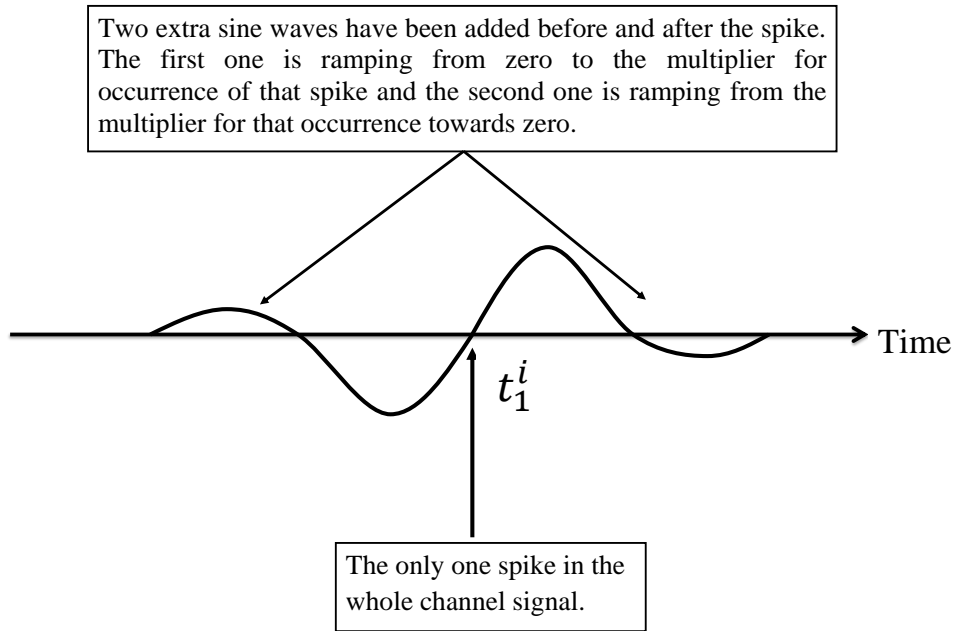


FIGURE 3.27: **A Single Spike:-** If there is only one spike in a particular channel, trivially two sine waves are created and concatenated together at the occurrence of the spike. The ramp technique is also applied in this case, to reduce the sharpness in the reconstructed signal.

3.8 Processing Time

The spike construction code has been amended so that the processing time involved in it can be minimized, as explained in section 3.2. The spikes have been coded in $Q(i)$ format rather than $P(i, j)$. This saves time to decode as the spikes are not necessary to be assigned in $Q(i)$ format.

As mentioned at [3], the spike occurrence time values have been assigned by each sensitivity level to find out sound features in the spike code. But that is not the most convenient way to code spike occurrence for the sake of efficiency of sound reconstruction. So, a new methodology (as described in MATLAB code in Appendix B) has been adopted where the spike occurrence time values have been saved according to each channel, not sensitivity level; as the reconstruction work considers spikes for each channel. This adaptation increases the construction and reconstruction work by much more efficiency and less processing time.

Constructing the spike code from a 2 sec long and 177 KB of size sound file, it takes 56 sec in the original spike coding technique to code spikes (figure 3.28) and 150 sec to decode those spikes (figure 3.29). But under the new technique, it takes only 7 seconds to code (figure 3.30) and 4 seconds to decode (figure 3.31). The system specification was Intel(R) Core(TM) i7 - 2600 CPU @ 3.40 GHz; 16 GB RAM; 64 bit operating system; MATLAB R2014a.

So previously under the old technique, it took $(56 + 150) = 206$ seconds to code the sound but only $(7 + 4) = 11$ seconds under the new technique. So, the processing time of coding and decoding has been reduced by 94.66%

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
AN_Construct_inNewSpikeForm_MP	1	55.654 s	0.060 s	
AN_1_SigGen_Mono_EditMP	1	55.594 s	1.714 s	
...ikeGen_Mono_MJN_inNewSpikeForm_EditMP	1	52.430 s	51.188 s	
sortrows	800	1.179 s	0.324 s	
gammatone1	50	0.991 s	0.688 s	
sortrows>sort_back_to_front	800	0.855 s	0.855 s	

FIGURE 3.28: **Original Technique:** Construction time of spike code of a sound.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
AN_Reconstruct	1	149.959 s	2.184 s	
AN_Reconstruct_AssignData_eachChannel	1	146.109 s	0.339 s	
AN_FindData	800	145.074 s	145.074 s	
...econstruct_GenerateSignal_eachChannel	1	1.538 s	1.445 s	
setdiff	704	0.549 s	0.047 s	

FIGURE 3.29: **Original Technique:** Re-construction time of that sound from its spike code.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
AN_Construct_inNewSpikeForm_MP	1	6.847 s	0.012 s	
AN_1_SigGen_Mono_EditMP	1	6.835 s	0.375 s	
...ikeGen_Mono_MJN_inNewSpikeForm_EditMP	1	5.525 s	5.415 s	
gammatone1	50	0.770 s	0.499 s	
unwrap	50	0.237 s	0.082 s	
waitbar	102	0.220 s	0.088 s	
unwrap>LocalUnwrap	50	0.151 s	0.151 s	

FIGURE 3.30: **New Technique:** Construction time of spike code of a sound.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
AN_Reconstruct	1	3.462 s	1.971 s	
...econstruct_GenerateSignal_eachChannel	1	1.377 s	1.268 s	
waitbar	102	0.194 s	0.072 s	
allchild	100	0.044 s	0.019 s	
waitbar>updateWaitbar	100	0.035 s	0.002 s	
uitools\private\uiwaitbar	102	0.034 s	0.034 s	

FIGURE 3.31: **New Technique:** Re-construction time of that sound from its spike code.

3.9 Channel-to-Channel Comparison of Decoding Technique

3.9.1 Detailed Comparison of various Reconstructed Sound

This biologically inspired spike based system is a lossy coding technique. So, it cannot be expected that the reconstructed sound signal will sound exactly the same like the original sound signal. By comparing channel to channel original and decoded signals, we will explain why this is the case.

Varying the number of channels and sensitivity levels and implementing various techniques alters the reconstructed sounds sounds. Again, one important fact must be considered that the original sound is passed through the GMF and the filterbank produce different channels with the center frequency varying from 100 Hz to $\min(\frac{f_s}{4}, 10kHz)$, where f_s is the sampling rate. The human ear is not very sensitive for the frequency higher than 10 kHz and lower than 100 Hz. So, the extra frequencies and energy present in those extreme frequencies will be omitted by the GMF.

To listen to the sounds, please open folder ‘Sound Files Under Test/Speech/Test File (My Name)’ in the attached DVD and choose any of the ‘.wav’ sounds.

Figure 3.32 and figure 3.33 compares the original bandpassed signal and the reconstructed signal for channel 1 i.e. 100 Hz. Those two figures show that the energy has been scattered for the decoded signal. This is true for the higher frequency channels (for example channel 45 i.e. 6974 Hz) which is shown by figure 3.34 and figure 3.35.

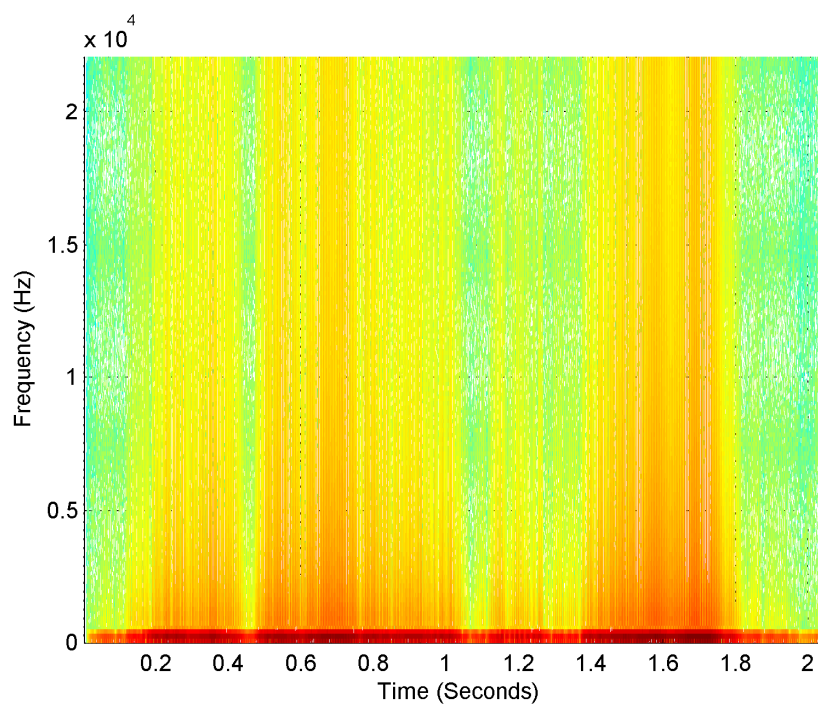


FIGURE 3.32: Spectrogram of the sound signal (directly from the GMF) for the lowest channel frequency.

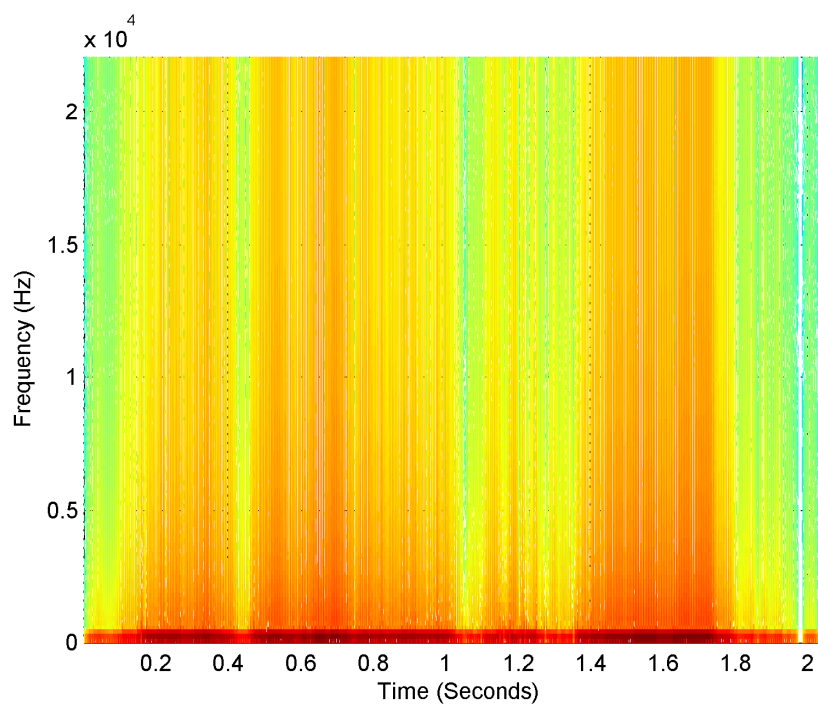


FIGURE 3.33: Spectrogram of the final reconstructed sound signal (reconstructed from spike coding) for that lowest frequency channel mentioned in figure 3.32. In comparison with the previous figure, there are lots of extra energy contents for that single channel signal. This happens because the spike coding technique is a lossy coding technique and that's why the reconstructed signal cannot sound exactly as the original sound.

The reconstructed sound adds harmonics, even if they have been minimized.

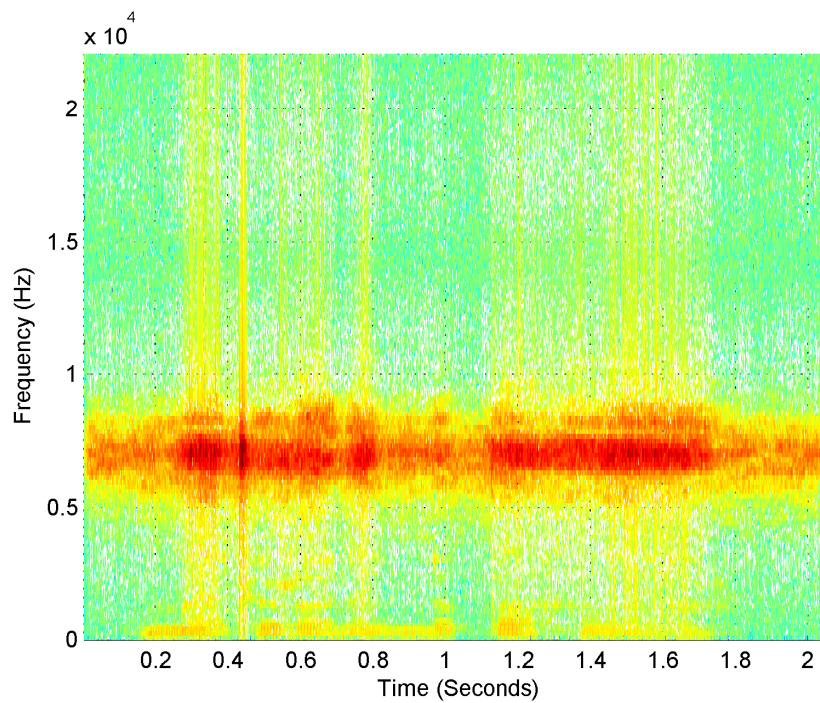


FIGURE 3.34: Spectrogram of the sound signal (directly from the GMF) for a high frequency channel. We can see that the energy contents tend to concentrate to the center frequency of GMF for that channel.

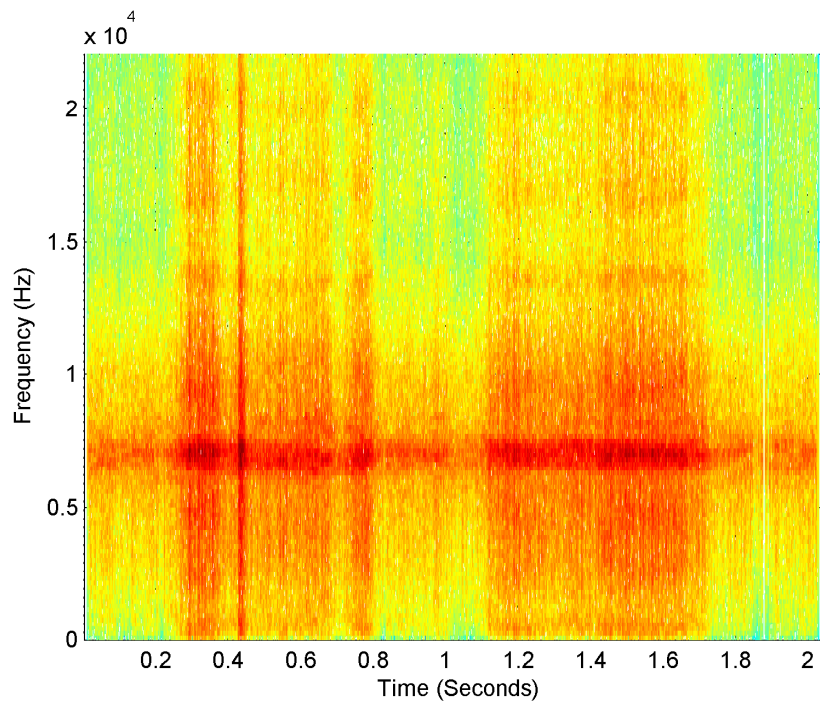


FIGURE 3.35: Spectrogram of the reconstructed sound signal (reconstructed from spike coding) for that high frequency channel mentioned in figure 3.34. In comparison with the previous figure, the energy contents scatter far away from the GMF center frequency for that channel. That's why the reconstructed sound does not exactly sounds like original sound.

By comparing these figures: 3.32, 3.33, 3.34, 3.35, we can say that the energies tend to scatter from its GMF center frequency in the reconstructed signal. The reason is that the spike coding technique is lossy and the spikes are not generated exactly at the GMF center frequency. So, when the sine waves are generated for each AN spike, its frequency deviates much from the center frequency of that channel. As the spike coding is a lossy coding technique, the reconstructed sound is never as good as the original sound.

3.10 Brief Summary of this Chapter

This chapter examines the AN spike coding technique efficiency. The full resynthesis algorithm from the AN spike code has been provided along with considering the delay vectors. The delay vectors were generated inside the gammatone filterbank and they are the delays of each bandpassed center frequency. Their effects on this resynthesis process has been described.

The number of channels and sensitivity levels has been changed to find out the optimum combination of those in producing the best quality of resynthesized sound. Our auditory nerve can fire spike at 200 spikes per second at the most. This fact has been implemented in coding spikes and then decoding sound from it and the quality of the resynthesized sound has been found as much worse than the previous one. Even jittering the spikes randomly didn't solve this issue so it has been concluded that due to the lossy nature of this coding technique, a quality of resynthesized sound is worse.

In this resynthesis process, some other techniques i.e. ramp technique, smoothing technique have been used and they have been well explained by figures. Also, the processing time of coding and decoding spikes have been considered and a better way to code and decode sound from it has been presented and it has been found that the processing time has been decreased by 94.66% by using the better coding technique. Lastly, the sounds have been compared on each individual channel (one low and one high frequency sound has been investigated here) and the resynthesized sounds will always be worse quality than the original sound. Clearly AN coding is lossy, but we are interested in reducing the perceptual-effect of these losses.

Chapter 4

Reconstructing Sound from Onset Spikes

In the previous chapter we have discussed how we have recreated sounds from AN spikes. Now we are going to discuss how onset spikes (see section 2.3.3) can be used to regenerate sound from its coded state. Generating the final sound involves reconstructing sound from three types of spikes -

1. **AN Spike:** which has been discussed previously;
2. **AN_Onset Spikes:** This is the amplitude modulated onset spikes which are more frequent than the original onset spikes & are introduced next;
3. **Original_Onset Spikes:** This is the original onset spikes as mentioned in ([3])

We have considered these three types of spikes, to recreate the sound to compare the quality of the reconstructed sound.

4.1 Amplitude Modulation

Amplitude modulation (AM) is a modulation technique used in electronic communication. In amplitude modulation, the amplitude (signal strength) of the carrier wave is varied in proportion to the waveform being transmitted [76]. AM has been used since 1800 for telegraph and telephone transmission [77].

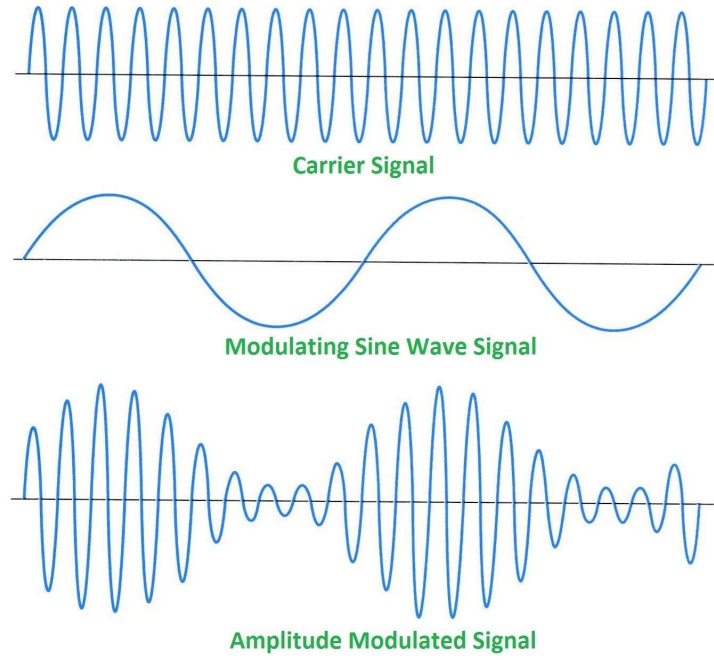


FIGURE 4.1: The top signal is the carrier signal (f_c^i) and the middle one is the modulated signal (f_m^i). The bottom one is the amplitude modulated signal modulated at f_m^i and carried by f_c^i , where $f_c^i > f_m^i$. Source: [78]

For each AN_Onset spike, an amplitude modulated signal has been generated and the AN_Onset spikes appear approximately 8 millisecond apart. So, the frequency of their appearance is $\frac{1}{0.008} = 125$ Hz. Note that 125 Hz is the approximate male speech's fundamental frequency. So, an amplitude modulated signal of 120 Hz, a little less than 125 Hz, can be used to see that is there an onset spike at the beginning of amplitude-rise in the signal. The carrier frequency has to be well above than 120 Hz so that the signal can be visually analyzed by the appearance of onset spikes and more than one harmonic is present in the GMF output. So, we have created a amplitude modulated signal modulated at 120 Hz and carried by 1500 Hz for each occurrence of AN_Onset spikes. The amplitude and the spectrogram of that signal has been shown in figure 4.2.

4.2 Introduction of AN_Onset Spikes

AN_Onset spikes are responsive to rapid increases that occur after a much shorter break than standard onset spikes. In that way, AN_Onset spikes are sensitive to AM. The original onset spikes usually appear about 100 milliseconds apart, but AN_Onset spikes

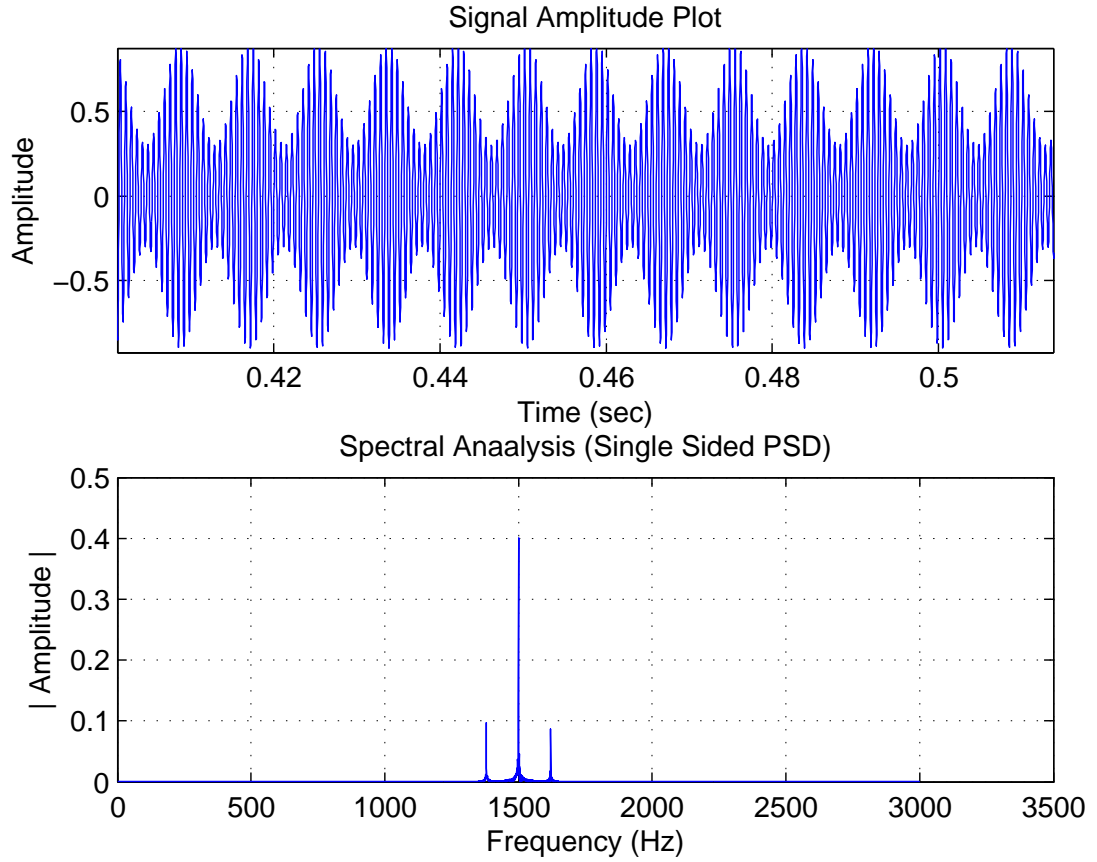


FIGURE 4.2: Amplitude and Spectrogram plot of the amplitude modulated signal, modulated at 120 Hz with a carrier frequency of 1500 Hz.

can appear about 8 milliseconds apart. They are able to detect amplitude modulations up to 125 Hz.

The reconstruction technique for onset spikes is to generate an amplitude modulated signal for each occurrence of an AN_Onset spike, unlike a sine wave for an AN spike. The onset generating technique should be able to produce onsets for each time the signal's energy has a increase i.e. it can be used as AM detector. If we cannot use our onset spikes to detect AM, then we cannot generate AM signal for each onset spike.

The GMF has the range of frequency 100 Hz to 10 kHz. Now for the middle frequency range (see section 4.5), AN_Onset spikes have been used. So, the reconstruction technique is that amplitude modulated signals were generated for each AN_Onset Spikes. The amplitude modulated signal modulated at 120 Hz and carried by the corresponding filterbank center frequency, has been constructed for each AN_Onset spike. For the original onset spikes, a short white noise has been generated. Figure 4.3 shows how the original onset spikes for different sensitivity levels appear.

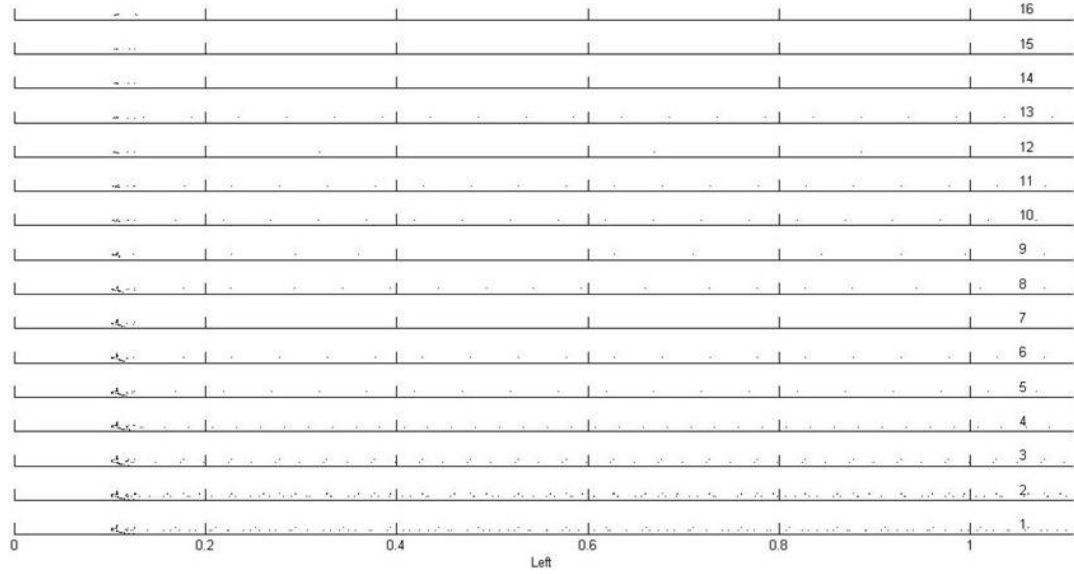


FIGURE 4.3: Here the x-axis represents time and each set of axes has the y-axis being the channel number (low to high frequency). Those sets are ordered from most sensitive to least sensitive. The first second of a 2 seconds long male speech has been used with the parameter values α , β and g : 100, 9 and 1100 respectively. The figure shows the appearances of original onset spikes (black dots) for different sensitivity levels for a speech signal. It can be seen that the onset spikes appear more in lower sensitivity levels but fewer in higher sensitivity levels.

It has been mentioned earlier in chapter 2 that the parameter α , β and g controls the onset generation technique in equations 2.5, 2.6 and 2.7. So, those parameter values have to be optimized first so that the onset can be generated in such a way that they appear at the beginning of each AM or at every increase of energy contents in a sound.

4.3 Modification of Onset Spike Generating Parameters

Appropriate parameter values are necessary to generate onset spikes at the right times. Initially the parameters α , β and g had the values 100, 9 and 1100 respectively. Figure 4.3 shows how those parameters create the onset spikes for different sensitivity levels. The figure 4.4 shows how the AN_Onset spikes appear for these parameter values.

The new values of the α , β and g has been chosen as 500, 25 and 1100 respectively. The AN_Onset spikes have been generated by using these parameters and they are appropriate. The onsets have been generated and they appear at the beginning of AM. They are shown in figure 4.4.

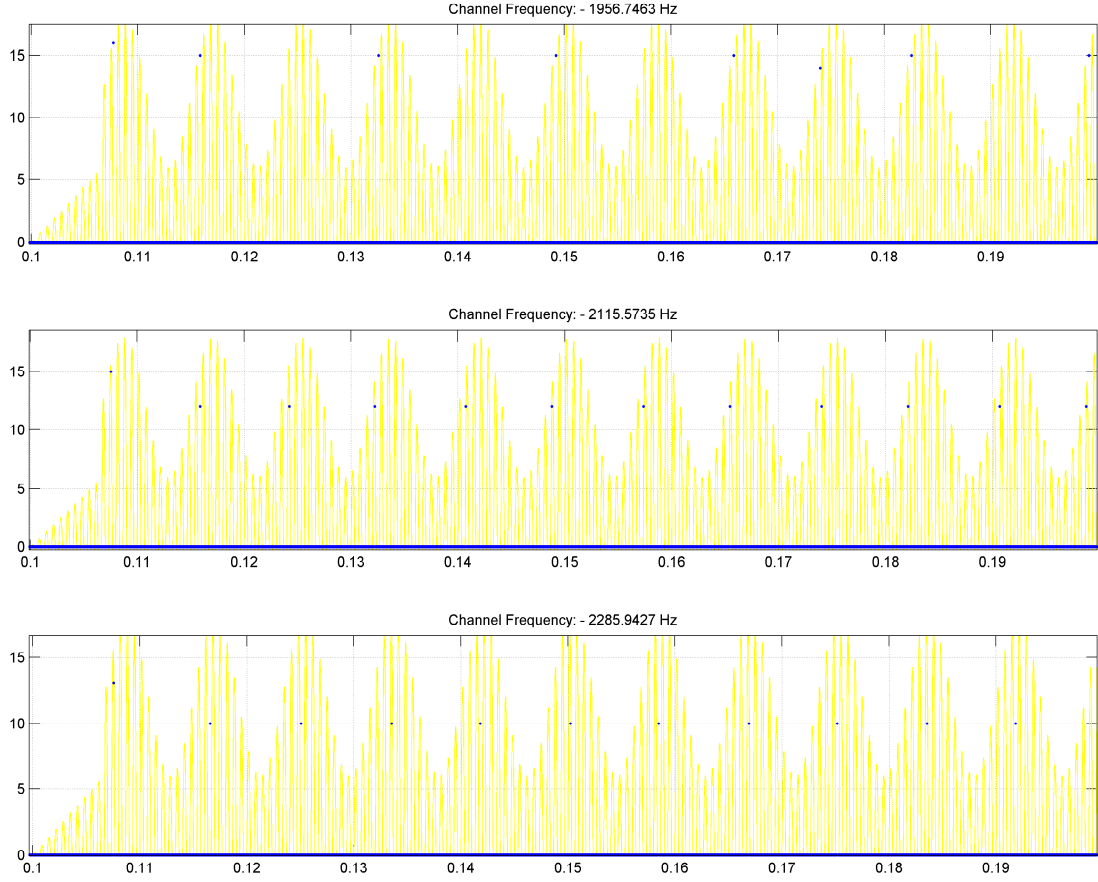


FIGURE 4.4: The figure shows that the onset spikes (blue dots) appear at appropriate times. Also, they are more in numbers, so the parameter values used to generate these onsets can be used for AM detector. Here x-axis is the time line and the y-axis has been the sensitivity levels in ascending order.

From figure 4.4, we can conclude that our onset generation technique almost perfectly detects AM. The AM detection technique has been successful by amending the parameters α , β and g . We therefore use the amended parameters for these sound reconstruction process.

Now our new values for the parameters α , β and g has been used to construct onset spikes in some real world sounds as well and they have been investigated to see the effectiveness of those parameter values. In this case **a clock tick-tock sound** has been considered, as this type of sound has a pause in the signal so there are good possibilities to see onsets at the right place. The AN_Onset spikes have been plotted together with the signal in the figure 4.5 to show that these AN_Onset spikes appear at the right times.

In figure 4.5, we can see the appearances of onset spikes which are exactly where we expect them to be. They appear at the beginning of each amplitude rise of the signal. Interestingly we can see that there are two consecutive onset spikes appear very near to

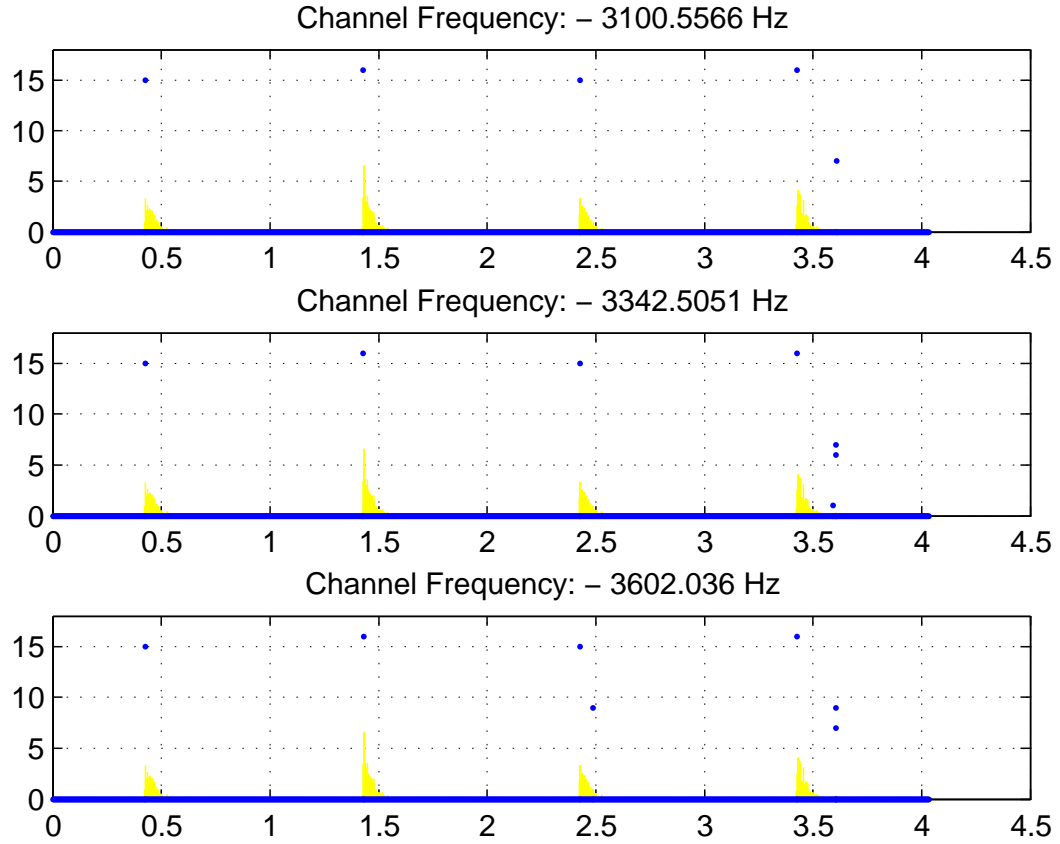


FIGURE 4.5: The figure shows that the onset spikes (in blue dots) have appeared at each ‘tick-tock’ sound of the clock. The appearance of these onset spikes is very accurate as well. Here x-axis is the time line and the y-axis has been the sensitivity levels in ascending order.

each other in figure 4.5. This happens because our onset detection technique measures the rise in the energy of the signal. Two consecutive rises in the energy spectrum of the signal may well produce two consecutive onset spikes although they does not really represent two different significant rise in the signal energy. However, this fact does not affect our reconstruction at all as we have used a filter to find a minimum distance, which is 8 milliseconds, between two consecutive onset spikes. We can conclude here that our onset detection technique is well prepared to produce onset spikes which can be used as AM detection. Now we are ready to regenerate the sound from these AN, AN_Onset and original onset spikes. Next the regeneration techniques have been described.

4.4 Three Major Types of Combination of AN and Onsets Spikes

4.4.1 A brief introduction

We have generated the original onset spikes with a quiet period of about 100 millisecond i.e. there is at least 100 millisecond (ms) of time gap between two consecutive onset spikes. However the AN_Onset spikes are much more frequent than the original onsets, having a quiet period of only 8 ms. The aim of these AN_Onset spikes is to recreate the amplitude modulated signals characteristic of voiced speech, but using a minimum number of spikes to code it.

Thus, the AN_Onsets appear more frequently than original onsets, for the sounds with shorter pauses. This does not happen with the sound which has several long pauses in it. To reconstruct sound, AN spikes, AN_Onset spikes and original onset spikes have been used at different frequencies as explained in table 4.1. For the lower frequency bands I_L , which has a range of about 0 - 1 kHz & where very few harmonics appear, AN spikes have been used. For the middle frequency of range I_M , 1 - 5 kHz, AN_Onset spikes have been used. At this frequency range, the bandwidth of the GMF is such for voiced speech that the output contains a number of adjacent harmonics of the fundamental, resulting in AM. However the middle range of frequency for male and female is different and discussed in detail at section 4.5. For the high frequency range I_H , say from 5 - 10 kHz where most of the sibilance ¹ are, the simply the original onsets have been used. For each occurrence of AN spikes, a sine wave has been generated. For the middle frequency range, an amplitude modulated signal has been generated for each occurrence of AN_Onset spikes. The amplitude modulated signal has the modulated frequency 125 Hz and has been carried by the relative GMF channel frequency. For the high frequency level, a short period of white noise has been generated for each original onset spike.

Why are we using this method to reconstruct the sound? The main reason is that our human ear is much more sensitive for the lower frequencies rather than higher frequencies. The most sensitive part of our ear is lower frequencies, where only the Auditory Nerve (AN) spikes have been used, as they are less lossy than the Onset

¹Sibilance is where strongly stressed consonants are made by producing air from vocal cords by deliberately using tongue and lips. This makes a hissing sound. [79]

spikes. For the middle part of frequency range, we have used AN_Onset spikes, which are less lossy than the original onset spikes as they are much more frequent. For the original onset spikes at higher frequencies, we have generated just white noise. As the human ear is not very sensitive at higher frequency, this white noise will not make the sound significantly worse than the original. Also, at the end, we have normalized the reconstructed signal for each type of frequency range. For the lower range it has been normalized by 0.9, making it the loudest. For the middle range, the signal has been normalized by 0.5 and the white noise for high frequency level has been normalized by 0.2. These have been discussed in detail later on in this chapter.

We have used AN, AN_Onset and original onset spikes to reconstruct the sound. The reconstruction of signals for each frequency channel from each type of spike is described below.

4.4.2 Reconstructing signal from *AN spikes*

The reconstruction of signal from the AN spikes is the same as reconstructing the sound from spike code at section 3.2. We had our AN spike as $\{P(i, j) : \text{where } i = 1, 2, \dots, N \text{ and } j = 1, 2, \dots, \xi\}$. The spike trains has been represented by a two dimensional sequences like: $P(i, j) = \{t_n^i\}$, where $t_n^i = (\hat{t}, \hat{j})$ for \hat{t} = time and \hat{j} = occurrence in highest sensitivity level of that spike with occurrences for sensitivity level $1 \leq j \leq \hat{j}$. We have generated the reconstructed signal for each frequency channel, similar with equation 3.8 as

$$s1_i^R(t) = \parallel_{k=1}^{K_i} s_k^i(t) \quad (4.1)$$

where, $\parallel_{i=1}^{K_i}$ means concatenating $s_1^i(t) \parallel s_2^i(t) \parallel \dots \parallel s_{K_i}^i(t)$. And $f_c^i = 100Hz$ to $600Hz$ or $1500Hz$, as we have used AN spikes for the Low Frequency Level (from 100 Hz to 600 Hz or 1500 Hz). After this, the signal has been normalized by 0.9

$$s1_i^{R'}(t) = 0.9 * s1_i^R(t) \quad (4.2)$$

4.4.3 Reconstructing signal from *AN_Onset spikes*

The difference between the original onsets and AN_Onsets is that the frequency of their appearance. The AN_Onsets are at minimum 8 ms apart, whereas the original onsets

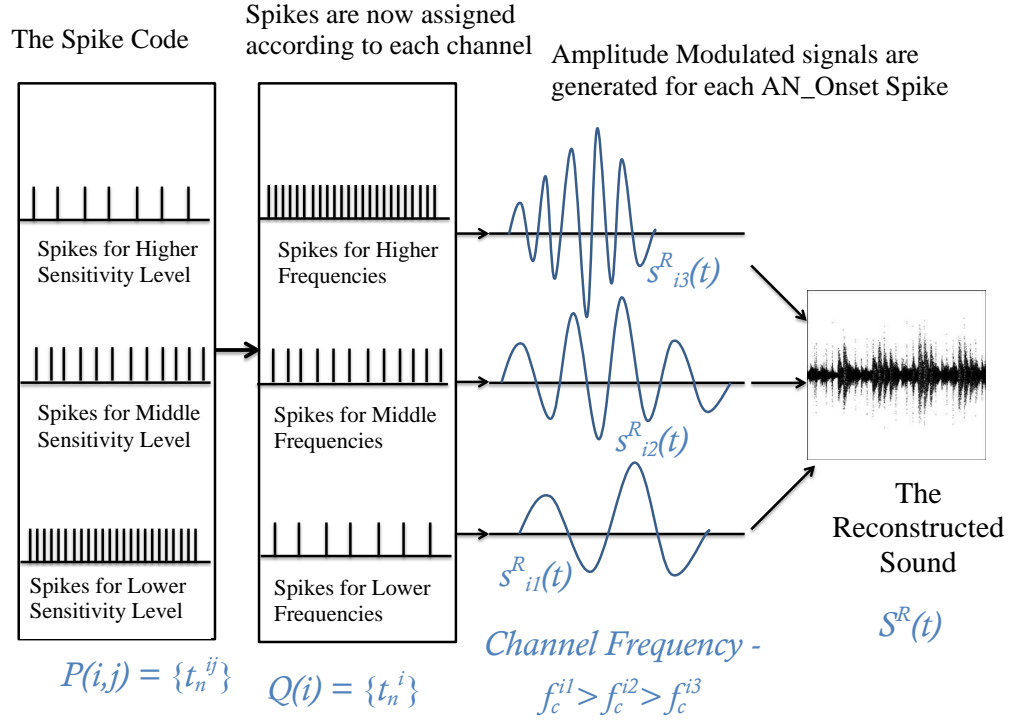


FIGURE 4.6: **The re-construction of sound from AN_Onset spikes:** The spike trains for each channel and sensitivity level are received from the AN spike code, where $P(i,j) = \{t_n^{ij}\}$, where i represents channels and j represents sensitivity levels. Then they are assigned according to each channel by combining their sensitivity levels and considering the highest sensitivity level. So, $P(i,j) = \{t_n^{ij}\}$, becomes $Q(i) = \{t_n^i\}$, where $t_n^i = (\hat{t}, \hat{j})$ for $\hat{t} = \text{time}$ and $\hat{j} = \text{occurrence in highest sensitivity level of that spike with occurrences for sensitivity level } 1 \leq \hat{j} \leq \xi$ (ξ is the total number of Sensitivity Levels). Then the amplitude modulated signals, modulated at 125 Hz and carried by the filterbank center frequency, are created for each channel according to each occurrence of spikes. At the end, the regenerated signals for each channels are summed up to get back the reconstructed sound signal $S^R(t)$.

are at least 100 ms apart in the signal. So, by using the AN_Onset spikes we can have more frequent onset spikes which are less lossy as they indicate more frequent rise in the energy spectrum of the signal. We have described before how can we use our onset spikes to detect the AM in the signal. So, for each occurrence of this frequent AN_Onset spikes we have similarly generated an amplitude modulated signal modulated at 120 Hz and carried by the respective filterbank channel frequency. The algorithm is similar to decoding AN spikes at section 3.2.

The Onset spikes can be assigned as $P(i,j)$, where $\{P(i,j): i = 1, 2, \dots, N \text{ \& } j = 1, 2, \dots, \xi\}$, which consists of all the spikes and their time values according to different sensitivity levels and different channels. Then, the spikes $P(i,j)$ has been assigned as $Q(i)$ like AN spikes. Here, $Q(i) = \{t_n^i\}$, where $t_n^i = (\hat{t}, \hat{j})$ for $\hat{t} = \text{time}$ and $\hat{j} = \text{occurrence}$

in highest sensitivity level of that spike with occurrences for sensitivity level $1 \leq \hat{j} \leq \xi$. So, here t_n^i and t_{n+1}^i are two adjacent onset spikes with both time value $\hat{t}_n^i, \hat{t}_{n+1}^i$ and occurrence in highest sensitivity level $\hat{j}_n^i, \hat{j}_{n+1}^i$ in i th channel. As mentioned in section 3.2, sorting spikes from $P(i, j)$ to $Q(i)$ [where i is the channel ($1 \leq i \leq N$) & j is the number of sensitivity levels ($1 \leq j \leq \xi$)], consumes much less time to decode the spikes. So, the onset spikes have been assigned in $Q(i)$ format.

The delay vectors, represented by D^i , where $i = 1, 2, \dots, N$ has been compensated by deducting D^i from the time values \hat{t}_n^i in $Q(i)$. So the delay compensated structure becomes $Q^d(i)$, like

$$Q^d(i) = \{\hat{t}_n^i - D^i\} \quad \text{for } i = 1, 2, \dots, N \quad (4.3)$$

The delay vectors have the same characteristics like $D^i \propto \frac{1}{i}$ (see [38, equation 41, page 69]) as explained in section 3.2. After $Q^d(i)$ becomes the sequences of delay compensated time values, amplitude modulated signals ($\hat{s}_k^i(t)$) have been generated between two consecutive AN_Onset spikes, t_{k-1}^i and t_k^i [where k is the occurrence of spikes].

The amplitude modulated signal has been modulated at frequency f_m^i and carried by frequency f_c^i (see figure 4.1). The AN_Onset spikes are minimum 8 millisecond. So,

$$f_m^i = \frac{1}{0.008} = 125Hz. \quad (4.4)$$

[where i is the channels ($1 \leq i \leq N$)]

$$f_c^i = \{\text{The center frequency of GMF at that channel}\} = f^i \quad (4.5)$$

Now, to generate the AM signals these frequencies f_m^i and f_c^i has been adjusted so that a perfect AM signal starting and ending at the value zero can be generated. So, for the modulated frequency a constant λ_m has been created where,

$$\lambda_m = (\hat{t}_k^i - \hat{t}_{k-1}^i) \times f_m^i \quad (4.6)$$

This λ_m has been rounded up as $\hat{\lambda}_m$ like:

$$\hat{\lambda}_m = \|\lambda_m\| \quad (4.7)$$

So, finally the adjusted modulated frequency becomes F_m^i and it can be described as:

$$F_m^i = \frac{\hat{\lambda}_m}{\hat{t}_k^i - \hat{t}_{k-1}^i} \quad (4.8)$$

Likewise, for the carrier frequency λ_c can be generated as

$$\lambda_c = (\hat{t}_k^i - \hat{t}_{k-1}^i) \times f_c^i \quad (4.9)$$

Then $\hat{\lambda}_c$ has been generated as

$$\hat{\lambda}_c = ||\lambda_c|| \quad (4.10)$$

And finally the adjusted carrier frequency becomes F_c^i and it can be explained as:

$$F_c^i = \frac{\hat{\lambda}_c}{\hat{t}_k^i - \hat{t}_{k-1}^i} \quad (4.11)$$

The amplitude modulated signal has been generated as $\hat{s}_k^i(t)$, where k denotes each occurrence of a spike, according to the following equation

$$\hat{s}_k^i(t) = 0.8 \sin(2\pi T F_c^i t) - 0.4 \sin(2\pi T (F_c^i - F_m^i) t) \quad (4.12)$$

where,

$$T = [\hat{t}_{k-1}^i + \frac{1}{R_{SM}}, \hat{t}_{k-1}^i + \frac{2}{R_{SM}}, \dots, \hat{t}_k^i] \quad \text{for } k = 2, 3, \dots, K_i \quad (4.13)$$

where R_{SM} is the sampling rate of the original sound signal.

Now, for each spike, there is an occurrence value attached to it. That value is the highest sensitivity level of occurrence of a spike. That tells how high the amplitude of the signal is between two consecutive spike occurring times. The value of the multiplier J_k^i (like in equation 3.4) for each AM signal is

$$J_k^i = \left(\frac{1}{\Theta_\xi}\right) \Theta_{\hat{j}} \quad \text{for } k = 1, 2, 3, \dots, K_i \quad (4.14)$$

where, Θ is the threshold Levels obtained from spike codes. ξ is the number of sensitivity levels used in the spike based code. So, for example, if a spike occurs at sensitivity level 3, the multiplier will be $\frac{1}{0.0362} \times 0.004 = 0.0110$, where 0.0362 is the threshold for sensitivity

level 16 and 0.0004 is the threshold for sensitivity level 3 (total number of sensitivity levels used is 16).

At this step, $\hat{s}_k^i(t)$ is multiplied by the multiplier J_k^i to get the right amplitude. A ramp technique, which is described at equation 3.20 in chapter 3, has been implemented on multiplier. So, the signal \hat{s}_k^i should be multiplied by \bar{J}_k^i and the signals with right amplitudes $[s_k^i(t)]$ are -

$$s_k^i(t) = (\bar{J}_k^i)[\hat{s}_k^i(t)] \quad \text{for } k = 2, 3, \dots, K^i \quad (4.15)$$

where \bar{J}_k^i has been explained in equation 3.20 in chapter 3.

Now for the very first spike, a sine wave has been generated with the length as the time value between the first and second spike in that particular channel. That sine wave is added to the front of the first spike's occurrence. That signal becomes

$$s_1^i(t) = (\bar{J}_1^i)[\sin(2\pi T \frac{1}{\hat{t}_2^i - \hat{t}_1^i} t)] \quad (4.16)$$

where \bar{J}_1^i has been explained in equation 3.21 in chapter 3.

So, finally the mathematical formula to concatenate all these signals, similar with equation 3.8, can be constructed as

$$s2_i^R(t) = \parallel_{k=1}^{K_i} s_k^i(t) \quad (4.17)$$

where, $\parallel_{i=1}^{K_i}$ means concatenating $s_1^i(t) \parallel s_2^i(t) \parallel \dots \parallel s_{K_i}^i(t)$. And $i = 600Hz$ or $1500Hz$ to $5kHz$ or $7kHz$, as we have used AN spikes for the Middle Frequency Level (from 600 Hz or 1500 Hz to 5 kHz or 7 kHz).

$$s2_i^{R'}(t) = 0.5 * s2_i^R(t) \quad (4.18)$$

4.4.4 Reconstructing signal from *Original Onset spikes*

The original onsets are the onset spikes which were generated by the onset generator at the beginning with time gap of about 100 ms. These spikes have been used in the higher frequency levels as mentioned in section 4.5. The range of higher frequency has been 5

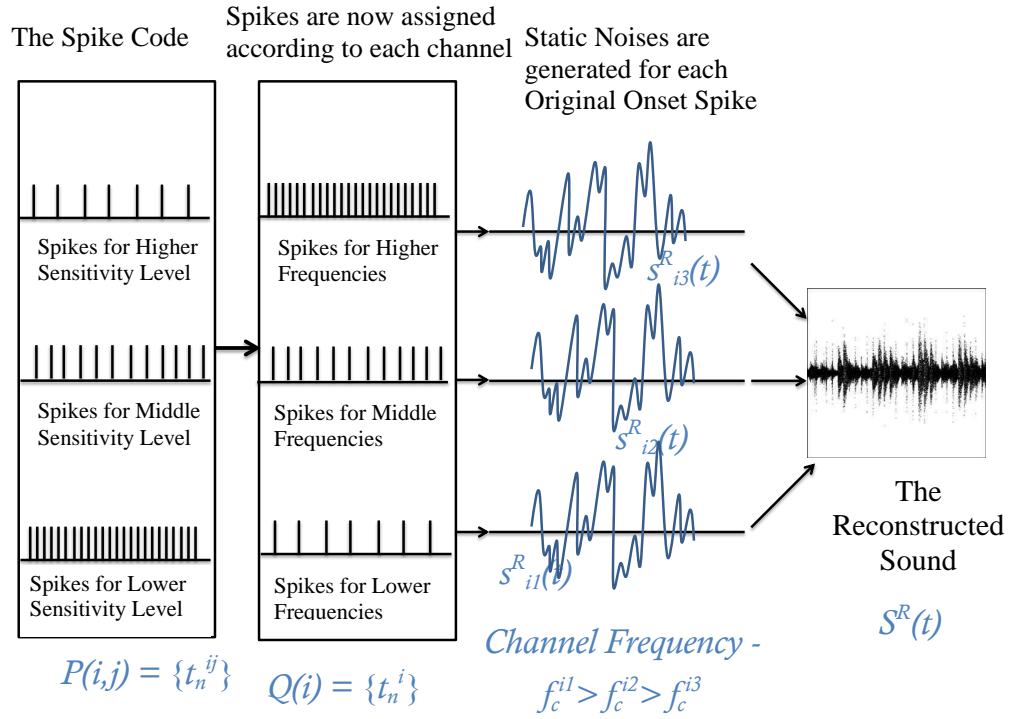


FIGURE 4.7: **The re-construction of sound from original onset spikes:** The spike trains for each channel and sensitivity level are received from the AN spike code, where $P(i,j) = \{t_n^{ij}\}$, where i represents channels and j represents sensitivity levels. Then they are assigned according to each channel by combining their sensitivity levels and considering the highest sensitivity level. So, $P(i,j) = \{t_n^{ij}\}$, becomes $Q(i) = \{t_n^i\}$, where $t_n^i = (\hat{t}, \hat{j})$ for $\hat{t} = \text{time}$ and $\hat{j} = \text{occurrence in highest sensitivity level of that spike with occurrences for sensitivity level } 1 \leq \hat{j} \leq \xi$ (ξ is the total number of Sensitivity Levels). Then the static white noises are created for each channel according to each occurrence of spikes. At the end, the regenerated signals for each channels are summed up to get back the reconstructed sound signal $S^R(t)$.

kHz or 7 kHz to 10 kHz. Our human brain is not very sensitive to hear high frequency sounds, so we have added some white (static) noise for each occurrence of these original onset spikes. We do not expect this will cause any noticeable decrease in the quality of reproduced sound, as those white noises will represent sibilance at high frequencies.

Like AN and AN_Onset spikes, these original onset spikes are generated in $P(i,j)$ form but later assigned in $Q(i)$ form. Then they are delay compensated as mentioned in equation 4.3 and becomes $Q^d(i)$. Here, $Q^d(i) = \{t_n^i\}$, where $t_n^i = (\hat{t}, \hat{j})$ for $\hat{t} = \text{time}$ and $\hat{j} = \text{occurrence in highest sensitivity level of that spike with occurrences for sensitivity level } 1 \leq \hat{j} \leq \xi$. So, it can be said that t_n^i and t_{n+1}^i are two adjacent spikes with both time value $\hat{t}_n^i, \hat{t}_{n+1}^i$ and occurrence in highest sensitivity level $\hat{j}_n^i, \hat{j}_{n+1}^i$ in i th channel. Now as the number of original onset spikes is very low, the time period between two adjacent spikes have been considered.

So, if two adjacent spikes have the time value $\hat{t}_{k-1}^i, \hat{t}_k^i$; τ_k^i can be used to denote the time gap between them as

$$\tau_k^i = \hat{t}_k^i - \hat{t}_{k-1}^i \quad \text{for } k = 2, 3, \dots, K^i \quad (4.19)$$

where K^i is the total number of spikes at the channel i .

If,

$$\tau_k^i > 3 \times \frac{1}{f^i} \quad (4.20)$$

where f^i is the center frequency of the GMF at channel i ; then the decoded signal $\hat{s}_k^i(t)$ is generated as:

$$\hat{s}_k^i(t) = \gamma^i(T_1 t) + \hat{\gamma}^i(T_2 t) + \gamma^i(T_3 t) \quad \text{for } k = 2, 3, \dots, K^i \quad (4.21)$$

where,

$$T_1 = [\hat{t}_{k-1}^i + \frac{1}{R_{SM}}, \hat{t}_{k-1}^i + \frac{2}{R_{SM}}, \dots, (\hat{t}_{k-1}^i + \frac{1}{f^i})] \quad \text{for } k = 2, 3, \dots, K^i \quad (4.22)$$

$$T_2 = [(\hat{t}_{k-1}^i + \frac{1}{f^i} + \frac{1}{R_{SM}}), \hat{t}_{k-1}^i + \frac{2}{R_{SM}}, \dots, (\hat{t}_k^i - \frac{1}{f^i})] \quad \text{for } k = 2, 3, \dots, K^i \quad (4.23)$$

$$T_3 = [(\hat{t}_k^i - \frac{1}{f^i} + \frac{1}{R_{SM}}), \hat{t}_{k-1}^i + \frac{2}{R_{SM}}, \dots, \hat{t}_k^i] \quad \text{for } k = 2, 3, \dots, K^i \quad (4.24)$$

where, $\gamma(t) \in \mathbb{N}$, randomly generated and $-1 \leq \gamma(t) \leq 1$ (represents white noise); R_{SM} is the sampling rate of the original sound signal & $\hat{\gamma}^i$ represents arrays of zeros.

If equation 4.20 does not hold, then the white noise $\hat{s}_k^i(t)$ has been generated like:

$$\hat{s}_k^i(t) = \gamma^i(T t) \quad (4.25)$$

where,

$$T = [\hat{t}_{k-1}^i + \frac{1}{R_{SM}}, \hat{t}_{k-1}^i + \frac{2}{R_{SM}}, \dots, \hat{t}_k^i] \quad \text{for } k = 2, 3, \dots, K^i \quad (4.26)$$

where, $\gamma(t) \in \mathbb{N}$, randomly generated and $-1 \leq \gamma(t) \leq 1$. where, R_{SM} is the sampling rate of the original sound signal.

The occurrence values of spikes can be obtained from equation 4.14 and after the multipliers have been multiplied with each $\hat{s}_k^i(t)$ according to equation 4.15 and $s_k^i(t)$ has

been obtained. A single sine wave has been added as well like in equation 4.16.

To obtain the final signal for each high frequency channel, we concatenate $s_k^i(t)$ like equation 3.8

$$s3_i^R(t) = \parallel_{k=1}^{K_i} s_k^i(t) \quad \text{for } k = 1, 2, 3, \dots, K^i \quad (4.27)$$

where, $\parallel_{i=1}^{K_i}$ means concatenating $s_1^i(t) \parallel s_2^i(t) \parallel \dots \parallel s_{K_i}^i(t)$. And $f_c^i = 5kHz$ or $7kHz$ to $10kHz$, as we have used AN spikes for the high frequency level (from 5 kHz or 7 kHz to 10 kHz).

$$s3_i^{R'}(t) = 0.2 * s3_i^R(t) \quad (4.28)$$

4.4.5 Summing up signals from *AN spikes*, *AN_Onset spikes* & *Original Onset spikes*

For all the GMF frequency ranges, the reconstructed signals has been generated, as $s1_i^{R'}(t)$, $s2_i^{R'}(t)$ & $s3_i^{R'}(t)$. At the end, these signals have been summed to achieve the final reconstructed signal $S^R(t)$ as

$$S^R(t) = \sum_{i=1}^N s1_i^{R'}(t) + s2_i^{R'}(t) + s3_i^{R'}(t) \quad (4.29)$$

Thus, $S^R(t)$ is the final decoded signal from the spikes.

4.5 Tuning the Decoding technique for Three Different types of Sound

This reconstruction work has been aimed for all sorts of sounds, but there are three major types among them. The GMF frequency range is from 100 Hz to 10 kHz. The frequency range has been divided into three sections as low frequency (I_L), middle frequency (I_M) and high frequency (I_H). Now these frequency ranges vary depending upon the type of sound. Largely there are three types of sounds:

	Low Frequency (I_L)	Middle Frequency (I_M)	High Frequency (I_H)
Male	100 Hz - 650 Hz	650Hz - 4843 Hz	4843 Hz - 10 kHz
Female	100 Hz - 1542 Hz	1542 Hz - 4843 Hz	4843 Hz - 10 kHz
Musical	100 Hz - 1542 Hz	1542 Hz - 6974 Hz	6974 Hz - 10 kHz

TABLE 4.1: The GMF frequency distribution for low, middle and high frequency range which have been used in reconstruction of sounds.

So, from this table 4.1, we can explain these:

- **Male:** Male voices have more energy in lower frequency bands than others. For Male, $I_L = 100$ Hz to 650 Hz, $I_M = 650$ Hz to 4843 Hz and $I_H = 4843$ Hz to 10 kHz.
- **Female:** Female voices have much more energy in the middle to high frequency than male voices. So, for female voices, $I_L = 100$ Hz to 1542 Hz, $I_M = 1542$ Hz to 4843 Hz and $I_H = 4843$ Hz to 10 kHz.
- **Musical:** There are other sorts of sounds, which has been classified as musical sounds. For the musical type of sound: $I_L = 100$ Hz to 1542 Hz, $I_M = 1542$ Hz to 6974 Hz and $I_H = 6974$ Hz to 10 kHz.

Our reconstruction technique does not automatically detect the type of sound. We have to specify it in the MATLAB function. The function-call in the MATLAB looks like `[AN_and_Onset_Reconstruct('filename_AN.mat', 'filename_AN_ANOnset.mat', 'filename_AN_OriginalOnset.mat', 3)]`. Notice at the end, we have used the number 3, which means that we have specified the type of the sound as musical and 'musical' type of reconstruction work has been done here. The number ('1', '2' and '3') represents 'Male', 'Female' and 'Musical' respectively. This has been mentioned in the Appendix A.

4.6 Issues in Sound Reconstruction from Onset Spikes

4.6.1 Normalization of decoded signals for different frequency levels

As it mentioned earlier, that for each type of spikes, different types of signal have been generated. So, for AN spikes, sine waves have been generated. For AN_Onset spikes, amplitude modulated signal has been generated and for the original onset spikes, simple white noise has been generated. Now among all these three types of spikes, AN spikes are least lossy and AN_Onset spikes are less lossy than original onset spikes as AN_Onset spikes are more frequent. So, to reflect this in the decoded signal, a multiplier has been added so that the decoded signal can contain least of the least-quality signals and most of the best quality signals. The multipliers are as follows:

- The sine waves created for AN spikes, has been multiplied by **0.9**, making it the loudest in the final decoded signal; according to equation 4.2.
- The amplitude modulated signals created for AN_Onset spikes have been multiplied by **0.5** making it reasonably loud in the final decoded signal; as mentioned in equation 4.18.
- White noises generated for original onset spikes, have been multiplied by **0.2** making it quietest in the final decoded signal; in equation 4.28.

So, the final signal containing all these signals will have loudest sine waves, louder AM signals and quiet white noises, as mentioned in equation 4.29.

4.6.2 Large time-gap between two adjacent spikes

Also, if there is a large gap between two spikes, we have generated decoded signals only where the spikes are. The algorithm has been explained in equation 4.20. As mentioned in equation 4.23, there is not any signals in between the signals for those two spikes. Figure 4.8 illustrates this.

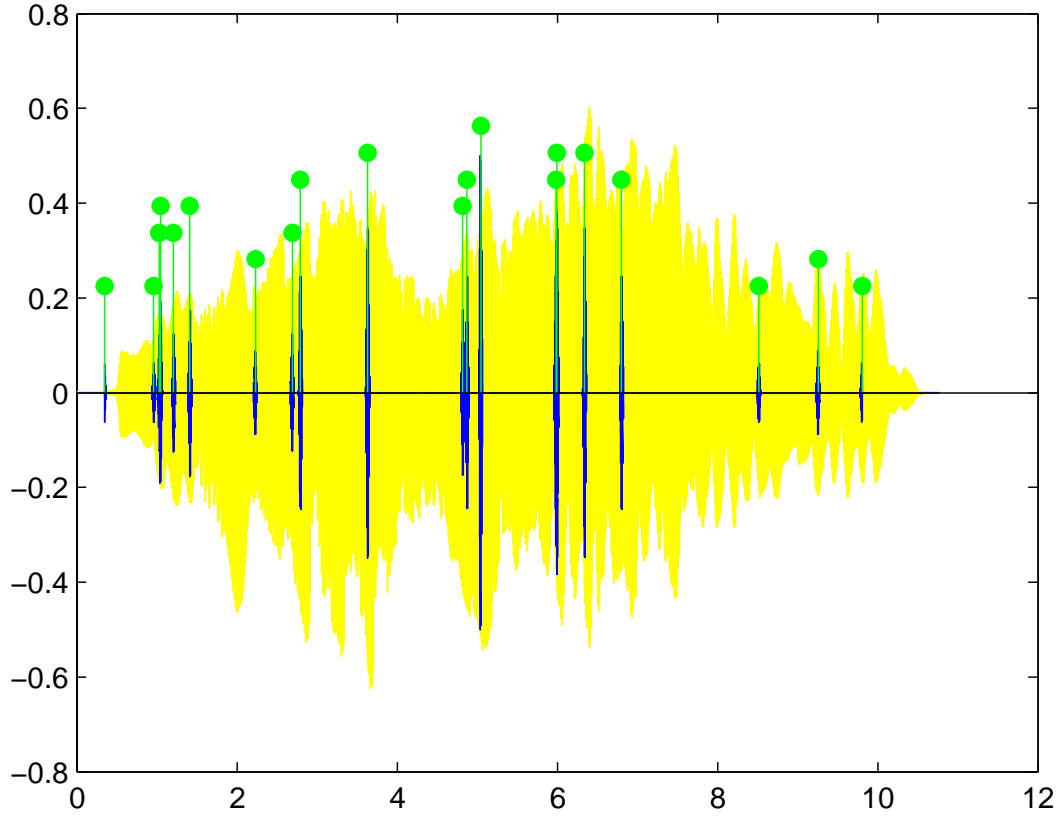


FIGURE 4.8: **Decoded signal generated for fewer spike:** This is the signal plot of a high frequency channel, where very few original onset spikes appear. The signal plot with yellow color at the background represents the original sound. The green straight lines appeared along the x-axis (which represents the sound length in second) are the occurrences of original onset spikes. Y-axis represents the amplitude values of the sound signal. The blue signals are the actual decoded signal generated for this high frequency channel. These blue sections show that the decoded signals are generated only where the spikes appear.

4.6.3 Effect of Delay vectors

Delay vectors have also been used in this onset spike reconstruction as in the AN spike reconstruction work. This delay vectors appear when the filterbank produce the signals for each channel and they delay for a certain period of time. The delay time has been subtracted from appearances of both AN and onset spikes for a particular channel frequency to reconstruct the sound according to equation 3.1.

4.6.4 Effect of Ramp Technique

Again like AN spike reconstruction, we have implemented ramp technique to avoid sudden jumps in the reconstructed signal. We have used the same kind of Ramp technique for both AN and Onset spikes according to equation 3.20, equation 3.21, equation 3.22.

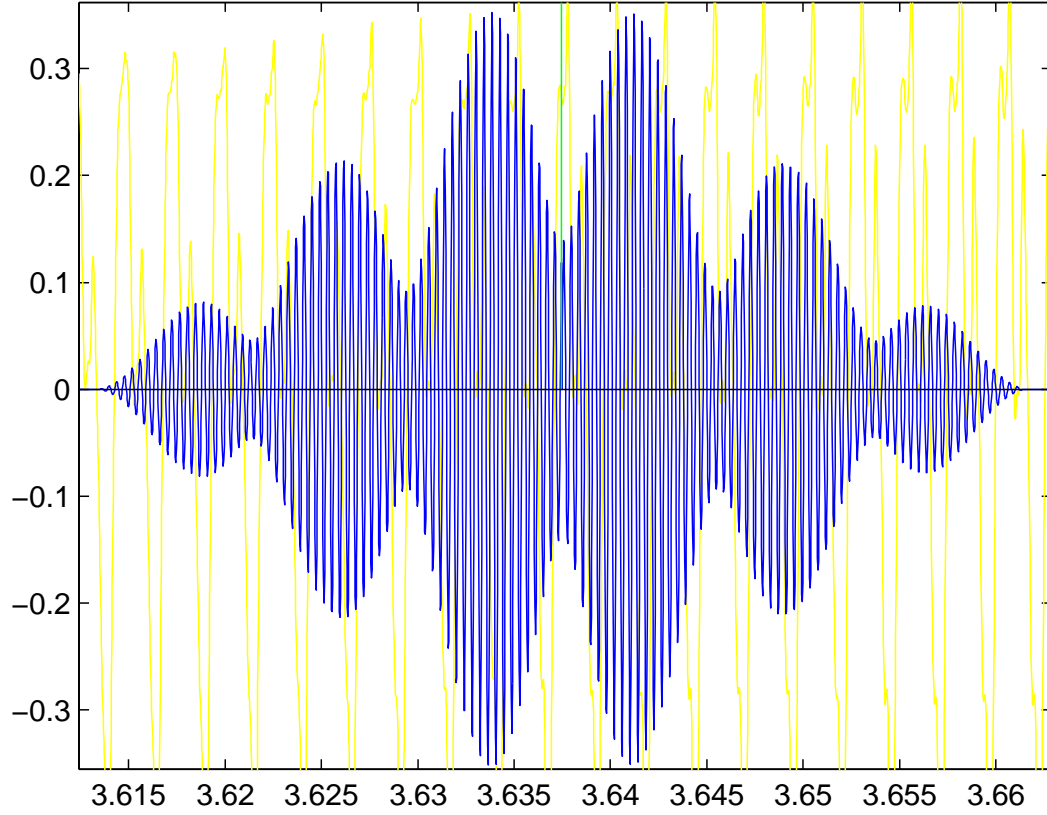


FIGURE 4.9: **Ramp Technique for onset spikes:** An Amplitude Modulated signal has been generated and then has been multiplied by a ramp multiplier as shown by the blue signals. In the background, the yellow signal is the original sound. X-axis represents the amplitudes and the Y-axis represents time in seconds.

4.7 Discussion on this Decoding Technique

The Onset spikes have been created for various types of sounds like: musical, choir, vocal and reverberated. These sound types are tested by comparing the decoded sound with the original in chapter 6.

The reconstruction work is definitely not producing decoded sounds as good as the original sounds. There are a few reasons for this including

- **Lossy Coding Technique:** Spike coding technique is lossy because we are losing information when we code a sound as spikes. When we regenerate the sound from its coded state, the quality of the decoded sound decreases. We have previously generated sounds from the AN spikes only and we have seen that the quality reduces but not by much. Now, onset spike code is much more lossy than the AN spikes. So, when reconstructing sounds from the onset spike codes, we cannot expect any better quality of reconstructed sound than AN spikes. So, the reason for the worse quality of the reconstructed sound can be explained on this general ground of lossy coding technique.
- **Our Reconstruction Techniques:** There exists some extra noises in the reconstructed signal in the higher frequency channels. The noise comes from the generated amplitude modulated signals for each occurrence of AN_Onset spikes. In reality AM modulated signals should not be generated for each AN_Onset spike, but we have used this decoding technique only for testing purpose. We were interested to see how good the reconstruction work can be if the AM signals can be used instead of the normal sine waves. That's why we got the noise at the middle-high frequency levels. Also, at the high frequencies we have only generated white noise for each onset spikes. But this does not affect our reconstruction work so much because the duration of white noise is really short and the amplitude of the noise has been multiplied by 0.2, so it does not appear to be loud.
- **Best Parameter Values:** The parameters used to generate the onset spikes are not at the most optimum level. The parameter values have been chosen when they were able to produce onset spikes at the desired places of the AM signals. Now, in reality this cannot be applied to all sorts of sounds in the natural world.

So, the lack of exact parameter values also contributes to the poorer quality of reconstructed sound. However, we have investigated the parameters, but believe they could be improved further.

4.8 De-coding from ‘Koickal’s Technique’

Koickal produced code to produce the decoded signal from his spike codes. It was designed for a hardware based system and works for analog signals. So, Koickal’s spike based technique reconstructs the sounds perfectly for artificial ramped sounds. But does not reconstruct the natural sounds like the sounds we hear every day at a very good quality. Koickal’s original code has been modified so that the decoded signal can be saved in the database and can be compared with the original signal. The code has been mentioned in the Appendix A. To generate the spikes, the resolution parameter has been set to 7 bits. This resolution parameters determines the bit depth of each spike. The PESQ testing in chapter 6 provides the evidence why this has been set to 7 bits rather than any other values like 5 or 9 bits.

Also, increasing this resolution parameter increases the number of generated spikes as described in the following table.

	5 bits	7 bits	9 bits
No of Spikes	113281	129933	147407

TABLE 4.2: The number of spikes increases with the increase in the bit depth for the Koickal coding of a male speech sound of 2 seconds in length.

4.9 Brief Summary of this Chapter

In this chapter, onset spikes have been used to decode the sound and there are two different types of onset spikes AN_Onset spikes and original onset spikes. AN_Onset spikes are more frequent than original onset spikes. To decode the sound, AN, AN_Onset and original onset spikes have been used and the decoding technique has been described in section 4.4. However to be able to use the onset spikes in decoding, the parameter values had to changed so that the onset spikes can appear at the right places.

We are using three different types of sounds - male, female and musical which have been classified by three different combination of frequencies as mentioned in table 4.1 followed by some other issues raised in this decoding. At the end it has been explained why the decoded sounds are not as good as the original sounds.

Koickal has decoded sounds from his spikes codes and his code uses the number of bits as a parameter. PESQ testing has led us to use 7 bits as explained in chapter 6.

Chapter 5

Subjective Testing & Findings

5.1 The Purpose of Testing

This research is involved with coding and decoding spike codes for sounds. The effectiveness of those spike codes can be compared by comparing the quality of decoded sounds. Testing is necessary to identify which spike coding technique produces better quality decoded sound. As discussed before, three major types of spike coding have been considered - AN spike code ([3]), Onset spike code ([4] and [3]) and Koickal's spike code ([5]). Subjective and objective testing of those reconstructed sounds can find out the quality of corresponding spike coding technique. If a sound coding technique produce spikes which can be decoded to good quality decoded sound, it can be claimed that sound coding technique is good. In this way, we can compare the effectiveness of those spike coding techniques.

This testing will also be marked for a proper testing for various spike coding technique. There have been number of tests about the quality and environments ([80]), but very few tests have been conducted on spike coding techniques. This testing will set up a benchmark for any other future spike based sound testing. Some journal papers provide helpful tips to carry out a proper sound testing. Among them, ([81] and [82]) have been used as the guidelines for the subjective testing, which has been conducted here.

5.2 Test Equipments Used

Subjective sound testing needs appropriate test equipments so that the sounds can be played properly and the participants can hear it clearly without any disturbance. Their responses should be filed properly so that they can be used for analysis purpose as well. Those subjective test equipments have been described below.

5.2.1 HTML & Audio Played in Web and Computer

Testing has been conducted on a computer where a web-page can be opened and the audios can be played back. The Web page has been created by using simple html (hypertext markup language). There are 20 questions to answer and for each question, a web page has been created. Each web page plays two sounds. The participant compares them and marks one of them ‘better than the other’. A sample question webpage along with html code has been provided in [Appendix C](#).

5.2.2 Soundproof Room and Headphones

This testing involves listening carefully to the sounds and then comparing them. So, an appropriate room and environment have been chosen. We use a sound proof chamber in that room has been used for the testing, to ensure that no other sounds can interfere.

Also, a good quality headphone (made by AKG, MkII, K240, 55 ohms) has been used to play the sounds to the participants. This way it has been made sure that the sound played in the web is absolutely clear to the audience.

5.2.3 Advertisements

An advertisement (see [Appendix C](#)) has been created to call out for volunteers. The participation in this testing was totally voluntary although some goodies and crisps were provided to the person after completing the test.

5.2.4 Volunteers

The number of volunteers is a crucial factor in this testing as there should be enough responses for each question so that a decision can be taken based on enough evidences. 21 volunteers have participated in this testing providing 21 answers which will be analyzed to find out which sound is better regarding that question. Each question compares the quality of two spike coding techniques; so based on those answers, the better spike coding technique can be suggested.

It can be noted here that the number of participants are low (only 21) in this testing. However, their responses have been very clear so more testers would not have made much difference in the final outcomes.

5.2.5 Forms

Each volunteer has to sign a ‘Consent Form’ which allows me to use their answers and responses and other information like age, gender in this PhD thesis. Next they are to fill in another ‘User Information Form’ which collects all of their information (specially hearing impairment or not) as data. An ‘Answer Script’ has been created so that the listeners can provide their choice in a written form (Appendix C).

5.2.6 Software

No special software has been used for this testing. However all of the data has been copied into Microsoft Excel for analysis purpose.

5.3 Test Procedure

This subjective test has 20 questions to answer and it is a very simple and straight forward test where the volunteer needs to do these following steps:

1. The volunteer needs to sit down and read the consent form and sign. Then he/she fills the information collector form and signs.

2. Then he/she puts the headphones on and the web site [here \(www.cs.stir.ac.uk/~mpa/testing/index.html\)](http://www.cs.stir.ac.uk/~mpa/testing/index.html) or 'www/testing/index.html' (in the attached DVD) is opened in front of him. The first web page tells the volunteer how to play the sound and write the correct answer down. Then he/she clicks on the Begin button to start the Testing.
3. The volunteer answers all 20 questions and at the end, he/she signs the answer scripts and concludes the Testing.
4. At the end, he/she is offered for some sweets and chips.

The entire testing takes about 12 to 15 minutes to complete. The volunteers are allowed to ask me any questions which they are not sure about. They are allowed to play the sounds as many times as they like and they are allowed to alter their answer.

The questions included the word 'better'. This caused a few confusions among the volunteers. It has been explained well to everybody that 'better' means whichever sound seems real and more realistic to them.

5.4 Background of Testing

5.4.1 Three Major Different Coding Techniques used in the Test

As mentioned before, we have considered three major spike coding in this research. They are summarized and very briefly restated here:

Technique 1: AN Spike Coding: This is the spike coding technique which has been described at section 2.3.2 in chapter 2. This technique is the proper implementation of auditory nerve actions and produces the largest number of spikes.

Technique 2: AN version of Onset Spike Coding: As mentioned before at section 2.3.3 in chapter 2, the difference between the original onset spikes and AN_Onset spikes is that the frequency of their appearance along the time domain. The AN_Onsets appear at minimum 8 ms apart, whereas the original onsets appear at minimum 100 ms apart in the time line of an input signal. So, by using the

AN Onset spikes, there are more frequent onset spikes which are less lossy as they indicate more frequent rise in the energy spectrum of the signal.

Technique 3: Koickal’s Spike Coding Technique: Koickal’s spike coding technique ([5]) has been described in section 2.3.4 at chapter 2 (Literature Review). We have reused his code for decoding.

5.4.2 Sounds used in Testing

In this subjective test, there are **six** different sounds which have been coded and then decoded from the spike codes. They all are from three different sound types, mentioned at section 4.5 in chapter 4. They are compared to each other in the questions in the Test. Four out of six are the musical sounds and others are male and female each. Out of four musical sounds, one is ‘percussion’ type of sound and other three are ‘string’ type of sound.

5.4.2.1 Celesta Sound (Frequency Level 4)

A Celesta sound at the frequency 4 has been chosen for the testing. This is a low frequency ‘**string**’ type of sound. It has been detailed in the figure 5.1.

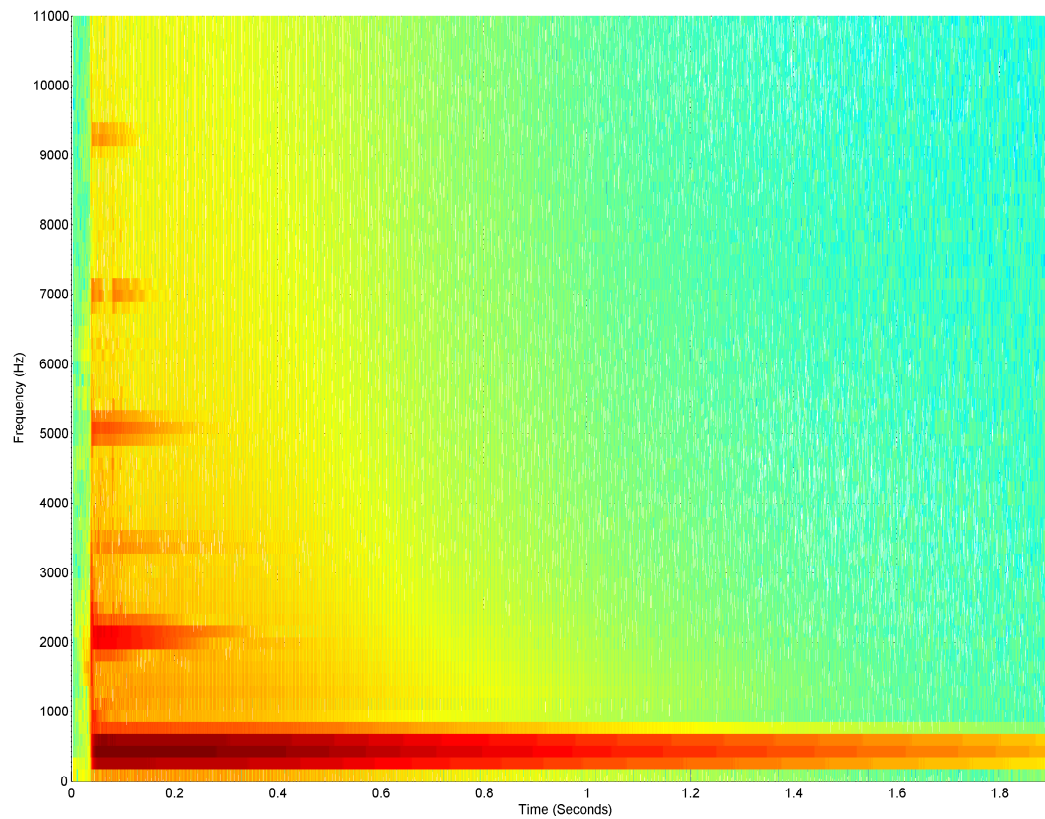


FIGURE 5.1: **Celesta sound (frequency level 4):** This spectrogram shows that the frequency contents are concentrated near 370 Hz. This sound can be heard by clicking [here](http://www.cs.stir.ac.uk/~mpa/testing/questions_1.html) (www.cs.stir.ac.uk/~mpa/testing/questions_1.html) and then by playing the FIRST sound.

5.4.2.2 Celesta Sound (Frequency Level 7)

Another Celesta sound has been used in the testing but at higher frequency level at 7. This is also ‘**string**’ type of sound. It has been detailed in the figure 5.2.

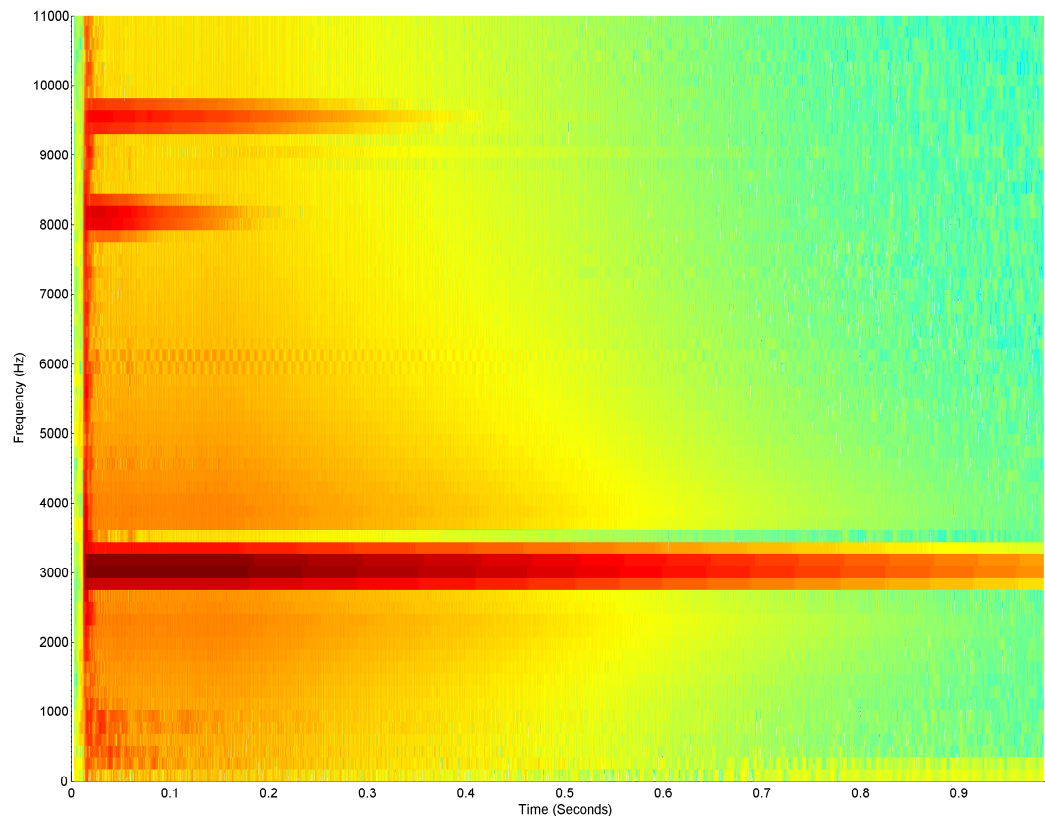


FIGURE 5.2: **Celesta sound (frequency level 7):** This spectrogram shows that the frequency contents are concentrated near 2986 Hz and apart from that there are very few frequency contents which are present in the spectrogram. This sound can be heard by clicking [here \(www.cs.stir.ac.uk/~mpa/testing/questions_2.html\)](http://www.cs.stir.ac.uk/~mpa/testing/questions_2.html) and then by playing the FIRST sound.

5.4.2.3 Electric Guitar Sound

An electric guitar sound has been considered in this testing as well. This is also ‘**string**’ type of sound. It has been detailed in the figure 5.3.

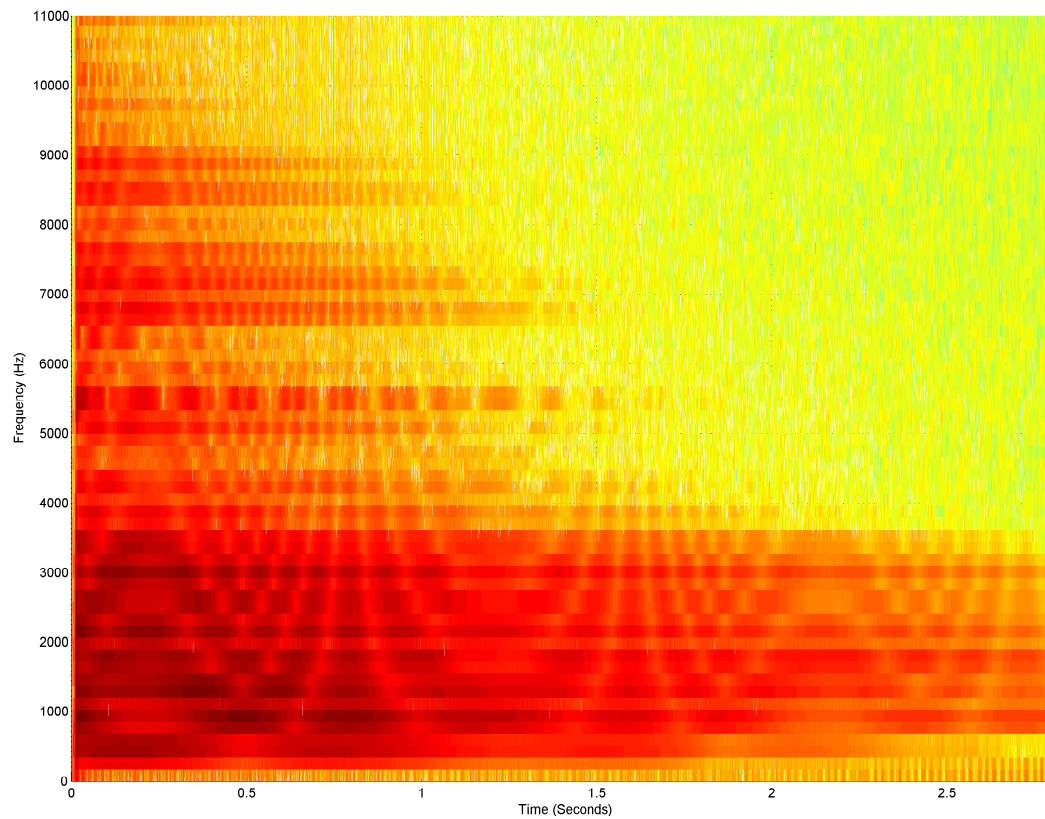


FIGURE 5.3: **Electric Guitar Sound:** This spectrogram shows that the frequency contents are distributed very densely from 500 Hz to about 3800 Hz. Apart from that there are few frequency contents which are present in the spectrogram. This sound can be heard by clicking [here](http://www.cs.stir.ac.uk/~mpa/testing/questions_16.html) (www.cs.stir.ac.uk/~mpa/testing/questions_16.html) and then by playing the FIRST sound.

5.4.2.4 Temple Bell

This is the sound of bell ringing in a temple. This is a ‘**percussion**’ type of sound. It has been detailed in the figure 5.4.

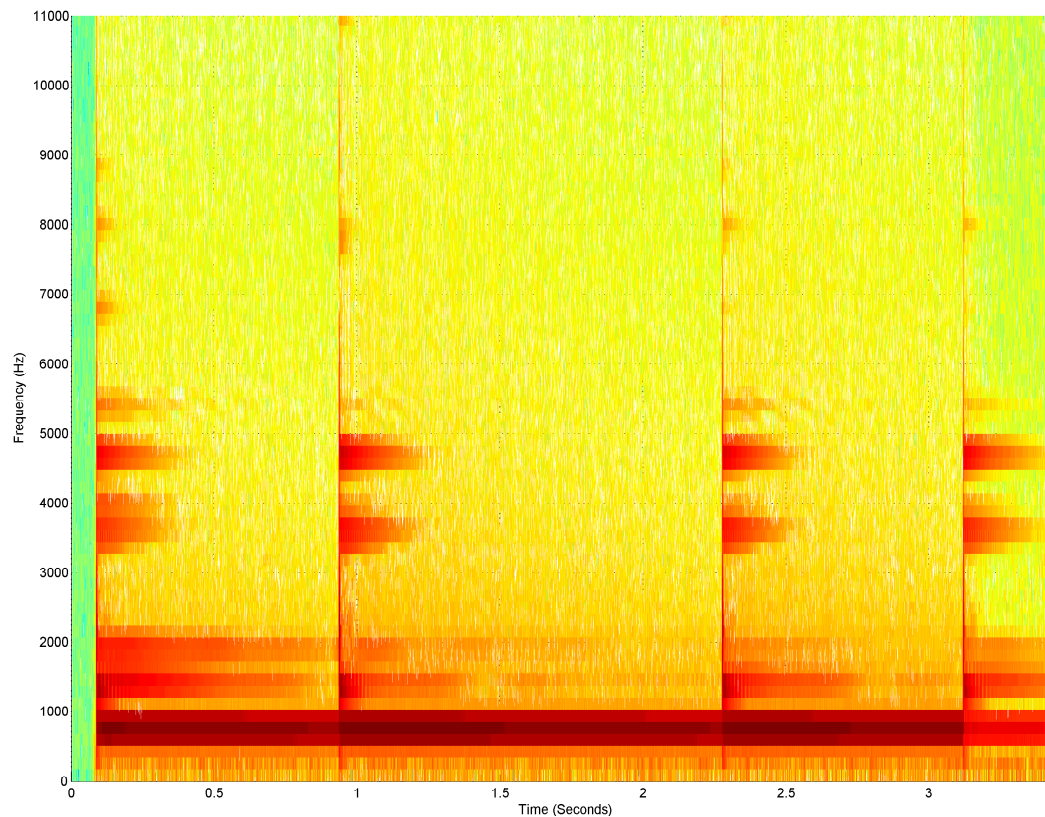


FIGURE 5.4: **Temple Bell:** This spectrogram shows that the frequency contents are distributed very densely from 500 Hz to about 1000 Hz. More frequency contents can be found at 3300 Hz to 4000 Hz and at 4500 Hz to 5000 Hz. This sound can be heard by clicking [here \(www.cs.stir.ac.uk/~mpa/testing/questions_3.html\)](http://www.cs.stir.ac.uk/~mpa/testing/questions_3.html) and then by playing the SEC-OND sound.

5.4.2.5 Male Voice

This is a male voice saying ‘Don’t ask me to carry an oily rag like that’ which has been included in this Testing. This is a ‘**male voice**’ type of sound taken from Timit dataset ([72]). It has been detailed in the figure 5.5.

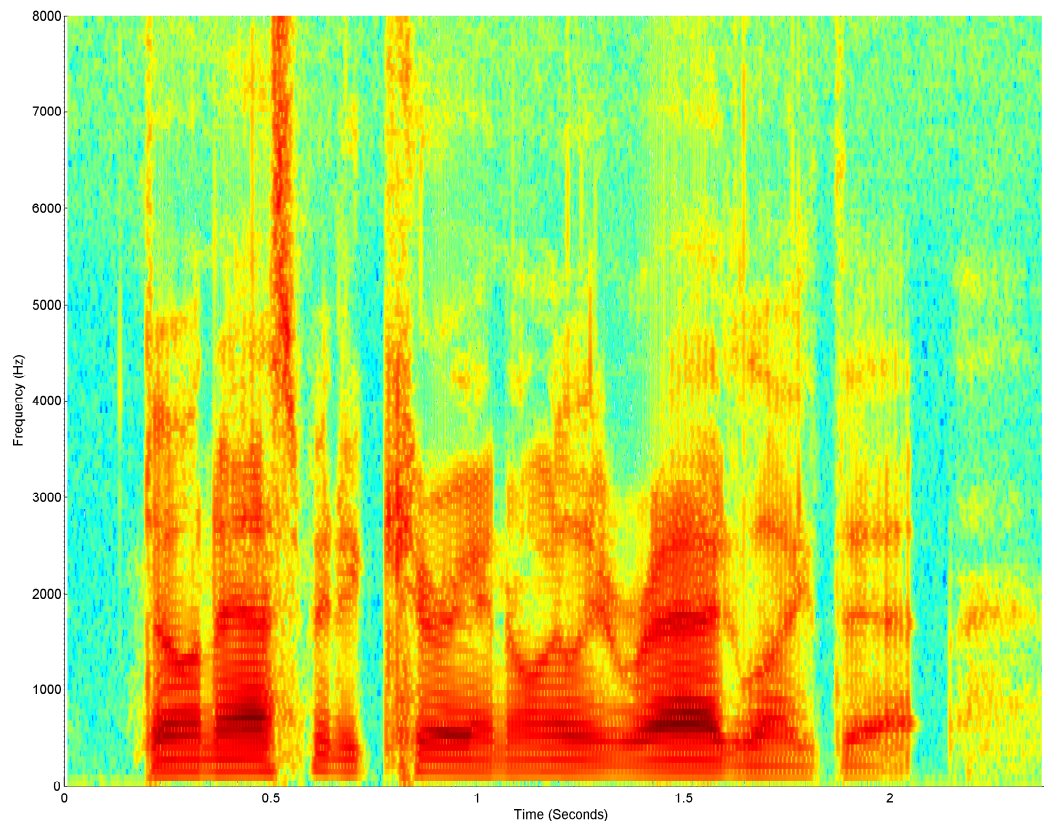


FIGURE 5.5: **Male voice:** This spectrogram of a male speech shows the typical frequency content distribution in a male speech signal. This sound can be heard by clicking [here](http://www.cs.stir.ac.uk/~mpa/testing/questions_10.html) (www.cs.stir.ac.uk/~mpa/testing/questions_10.html) and then by playing the FIRST sound.

5.4.2.6 Female voice

This is a female voice saying ‘Don’t ask me to carry an oily rag like that’ which has been included in this Testing. This is a ‘**female voice**’ type of sound taken from Timit dataset ([72]). It has been detailed in the figure 5.6.

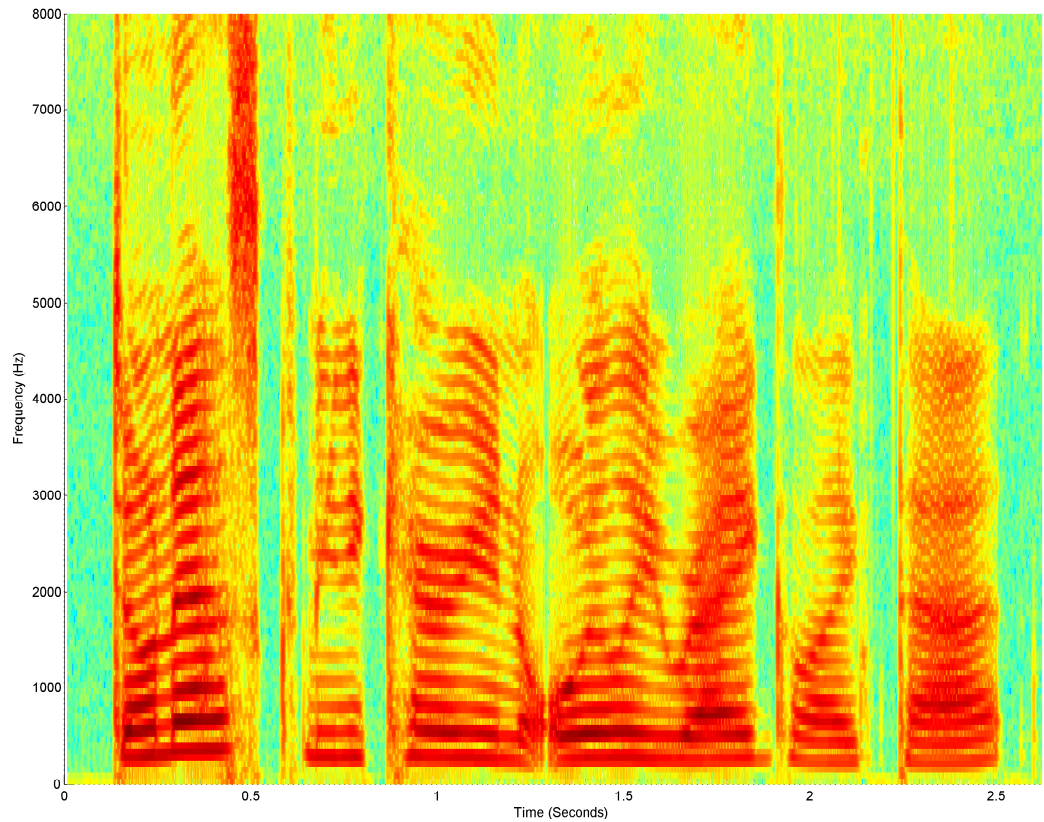


FIGURE 5.6: **Female Voice:** This spectrogram of a female speech shows the typical frequency content distribution in a female speech signal. This sound can be heard by clicking [here](http://www.cs.stir.ac.uk/~mpa/testing/questions.6.html) (www.cs.stir.ac.uk/~mpa/testing/questions.6.html) and then by playing the FIRST sound.

5.5 The Testing Questions and the techniques compared in each questions

Here each subsection will explain the type of techniques used to decode the sound and their corresponding details. The Test is accessible [here](http://www.cs.stir.ac.uk/~mpa/testing/): (www.cs.stir.ac.uk/~mpa/testing/). All 20 comparisons have been made by asking the question: ‘Which Sound is better?’, which is a ‘Two-alternative forced choice’ [83].

5.5.1 Question 1: Original Sound vs AN-Decoded Sound

Here, the celesta ‘string’ type frequency 4 level sound has been decoded from the AN spike code and then played and compared with the original sound. The first sound played is the original sound and the second one is the decoded from AN spike codes. Table 5.1 compares these two sounds in detail.

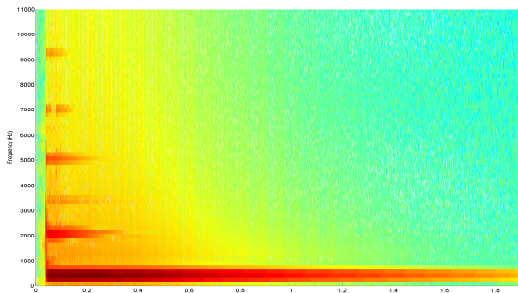
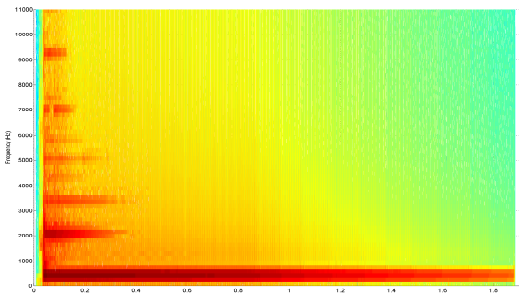
Question 1		
Coding Technique:	Original	AN Coding
Sound Type:	Celesta Frequency 4 (String Type)	
URL:	www.cs.stir.ac.uk/~mpa/testing/questions_1.html	
Spectrogram:	 <p>Spectrogram of the original sound</p>	 <p>Spectrogram shows that the energy is more scattered in the decoded signal.</p>
Test Answer (in favor):	11	10
Observed Probability:	0.5238	0.4762

TABLE 5.1: Test Question 1. Here 11 out of 21 have favored the original sound over the AN-decoded sound. The spectrograms are fairly similar to each other.

5.5.2 Question 2: Original Sound vs AN version of Onset-Decoded Sound

Here, the Celesta ‘string’ type frequency level 7 sound has been played. First the original sound has been played with the AN-Onset decoded sound. Table 5.2 compares these two sounds in detail.

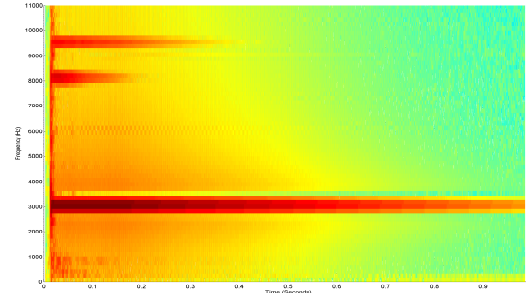
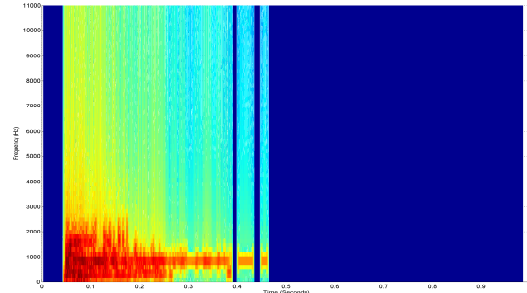
Question 2		
Coding Technique:	Original	AN version of Onset Coding
Sound Type:	Celesta Frequency 7 (String Type)	
URL:	www.cs.stir.ac.uk/~mpa/testing/questions_2.html	
Spectrogram:	 <p>Spectrogram of the original sound</p>	 <p>Spectrogram shows that the energy is missing in higher frequencies.</p>
Test Answer (in favor):	19	2
Observed Probability:	0.9048	0.0952

TABLE 5.2: Test Question 2. Here 19 out of 21 have favored the original sound over the AN_Onset-decoded sound. This is to be noticed here that the spectrograms of the original and decoded sound are very different from each other. This is because AN_Onset spike coding is not sensible at all for the high frequency sounds. Here a celesta high frequency level 7 has been chosen, so AN_Onset spike coding cannot contain high frequency contents in it.

5.5.3 Question 3: AN version of Onset-Decoded Sound vs Original

Here, the Temple Bell ‘percussion’ type of sound has been played. The AN-Onset decoded sound has been played first followed by the original sound next. Table 5.3 compares these two sounds in detail.

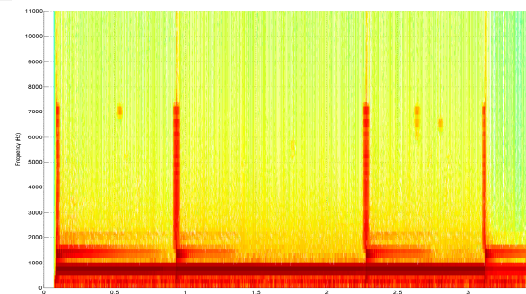
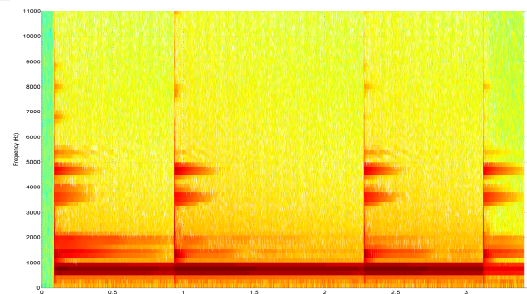
Question 3		
Coding Technique:	AN version of Onset Coding	Original
Sound Type:	Temple Bell (Percussion Type)	
URL:	www.cs.stir.ac.uk/~mpa/testing/questions_3.html	
Spectrogram:	 <p>Spectrogram of AN-Onset-Decoded sound shows missing energies in higher frequencies</p>	 <p>Spectrogram of the original signal.</p>
Test Answer (in favor):	1	20
Observed Probability:	0.0476	0.9524

TABLE 5.3: Test Question 3. Here only 1 out of 21 has favored the AN-Onset-decoded sound over the original sound.

5.5.4 Question 4: AN-Decoded Sound vs Original

Here, the Male ‘Speech’ type of sound has been played. The AN spike decoded sound has been played first followed by the original sound next. Table 5.4 compares these two sounds in detail.

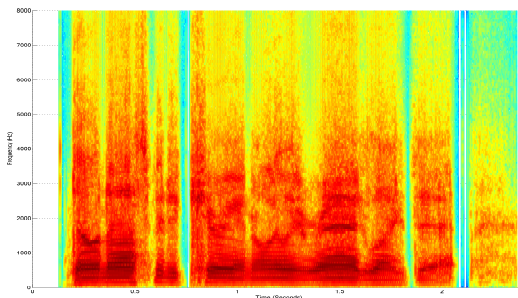
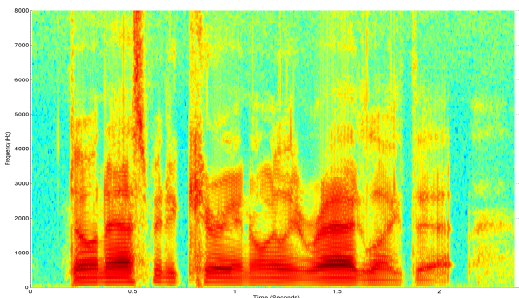
Question 4		
Coding Technique:	AN Coding	Original
Sound Type:	Male Speech (Speech Type)	
URL:	www.cs.stir.ac.uk/~mpa/testing/questions_4.html	
Spectrogram:	 <p>Spectrogram of AN-Decoded sound which shows that the energy is scattered everywhere and there are very little similarity with the original sound's spectrogram</p>	 <p>Spectrogram of the original signal.</p>
Test Answer (in favor):	0	21
Observed Probability:	0	1

TABLE 5.4: Test Question 4. Here nobody has favored the AN-decoded sound over the original sound.

5.5.5 Question 5: AN-Decoded Sound vs AN version of Onset-Decoded Sound

Here, the Temple Bell ‘percussion’ type of sound has been played. The AN spike decoded sound has been played first followed by AN-Onset-Decoded sound next. Table 5.5 compares these two sounds in detail.

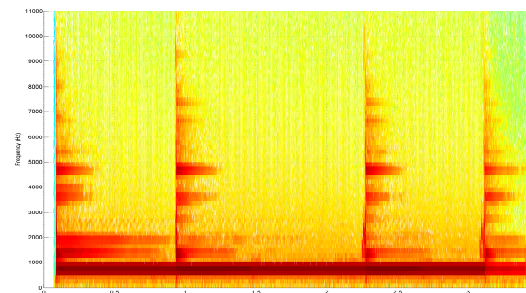
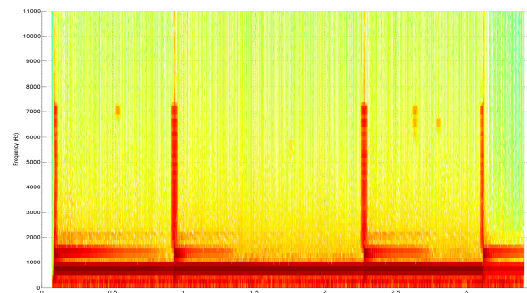
Question 5		
Coding Technique:	AN Coding	AN version of Onset Coding
Sound Type:	Temple Bell (Percussion Type)	
URL:	www.cs.stir.ac.uk/~mpa/testing/questions_5.html	
Spectrogram:	 <p>The spectrogram shows that energy is present for various frequency</p>	 <p>The spectrogram follows the other one but less energy is present in here.</p>
Test Answer (in favor):	19	2
Observed Probability:	0.9048	0.0952

TABLE 5.5: Test Question 5. Here 19 out of 21 have favored the AN-decoded sound over the AN_Onset-Decoded sound.

5.5.6 Question 6: Original vs AN version of Onset-Decoded Sound

Here, the Female ‘speech’ type of sound has been played. The original sound has been played first followed by AN-Onset-Decoded sound next. Table 5.6 compares these two sounds in detail.

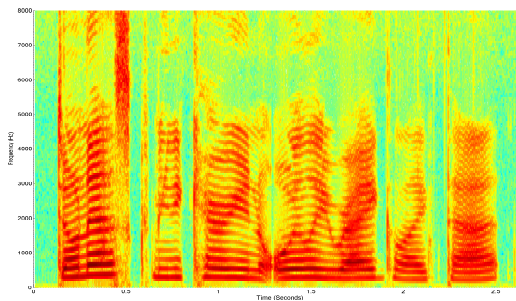
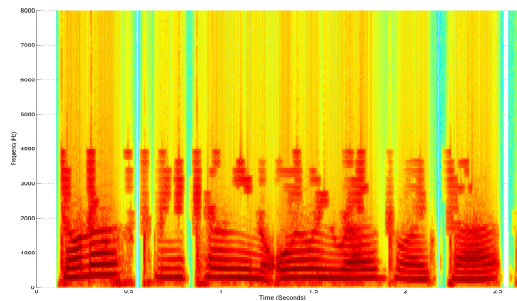
Question 6		
Coding Technique:	Original	AN version of Onset Coding
Sound Type:	Female (Speech Type)	
URL:	www.cs.stir.ac.uk/~mpa/testing/questions_6.html	
Spectrogram:	 <p>Spectrogram of the original sound</p>	 <p>Spectrogram shows the same pattern but the energy has been overly distributed.</p>
Test Answer (in favor):	21	0
Observed Probability:	1	0

TABLE 5.6: Test Question 6. Here all of the participants have favored the original sound over the AN-Onset-Decoded sound.

5.5.7 Question 7: AN-Decoded Sound vs Koickal's Reconstructed Sound

Here, the Celesta 'string' type frequency level 7 sound has been played. First the An-Decoded sound has been played followed by the Koickal's Reconstructed sound next.

Table 5.7 compares these two sounds in detail.

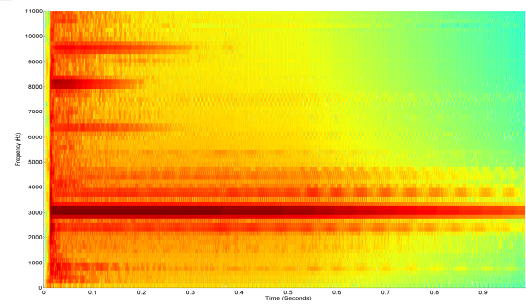
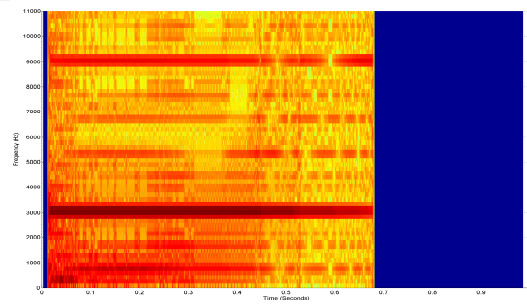
Question 7		
Coding Technique:	AN Coding	Koickal's Spike Coding
Sound Type:	Celesta Frequency 7 (String Type)	
URL:	www.cs.stir.ac.uk/~mpa/testing/questions_7.html	
Spectrogram:	 <p>The spectrogram of the AN-Decoded sound shows that the energy follows the right pattern</p>	 <p>Here the spectrogram shows that the energy has been overly distributed almost everywhere.</p>
Test Answer (in favor):	21	0
Observed Probability:	1	0

TABLE 5.7: Test Question 7. Here all of the participants have favored the AN-Decoded sound over the Koickal's Reconstructed sound.

5.5.8 Question 8: Original vs Koickal’s Reconstructed Sound

Here, the Celesta ‘string’ type frequency level 4 sound has been played. First the original sound has been played followed by the Koickal’s Reconstructed sound next. Table 5.8 compares these two sounds in detail.

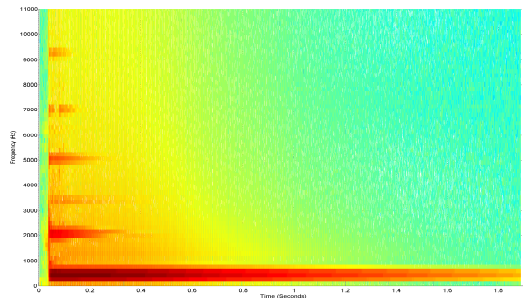
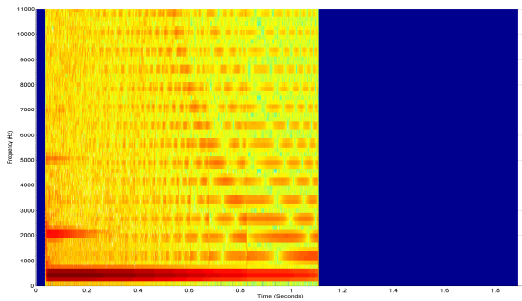
Question 8		
Coding Technique:	Original	Koickal’s Spike Coding
Sound Type:	Celesta Frequency 4 (String Type)	
URL:	www.cs.stir.ac.uk/~mpa/testing/questions_8.html	
Spectrogram:	<div><p>Spectrogram of the original sound</p></div>	<div><p>Here spectrogram shows that the energy has been overly distributed almost everywhere.</p></div>
Test Answer (in favor):	21	0
Observed Probability:	1	0

TABLE 5.8: Test Question 8. Here all of the participants have favored the AN-Decoded sound over the Koickal’s Reconstructed sound.

5.5.9 Question 9: AN version of Onset-Decoded Sound vs Koickal's Reconstructed Sound

Here, the Celesta 'string' type frequency level 7 sound has been played. First the An-Onset-Decoded sound has been played followed by the Koickal's Reconstructed sound next. Table 5.9 compares these two sounds in detail.

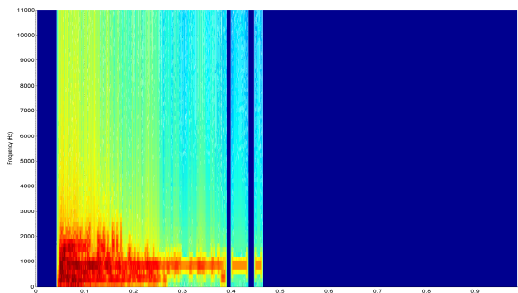
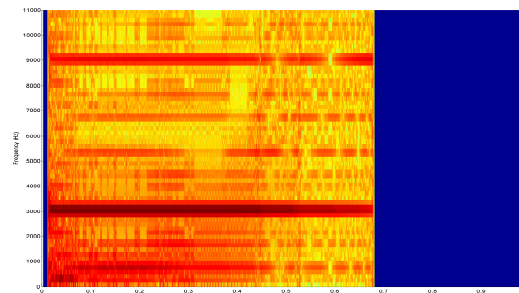
Question 9		
Coding Technique:	AN Version of Onset Coding	Koickal's Spike Coding
Sound Type:	Celesta Frequency 7 (String Type)	
URL:	www.cs.stir.ac.uk/~mpa/testing/questions_9.html	
Spectrogram:	 <p>The spectrogram can show the energy distribution only for a little while</p>	 <p>Here the spectrogram shows that the energy has been overly distributed almost everywhere.</p>
Test Answer (in favor):	16	5
Observed Probability:	0.7619	0.2381

TABLE 5.9: Test Question 9. Here 16 out of 21 have favored the AN-Onset-Decoded sound over the Koickal's Reconstructed sound. This is to be noticed here that the spectrograms of the AN_Onset decoded sound and Koickal's decoded sound are very different from each other, similar with question 2. This is because AN_Onset spike coding is not sensible at all for the high frequency sounds. Here a celesta high frequency level 7 has been chosen, so AN-Onset spike coding cannot contain high frequency contents in it.

5.5.10 Question 10: Original vs AN version of Onset-Decoded Sound

Here, the Male ‘Speech’ type sound has been played. First the original sound has been played followed by the AN-Onset-decoded sound next. Table 5.10 compares these two sounds in detail.

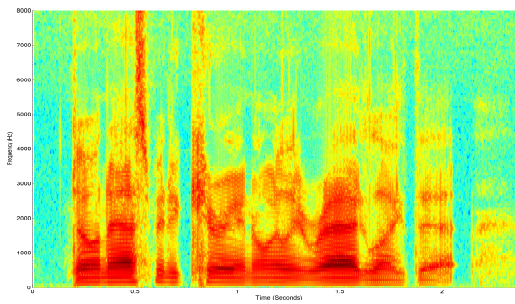
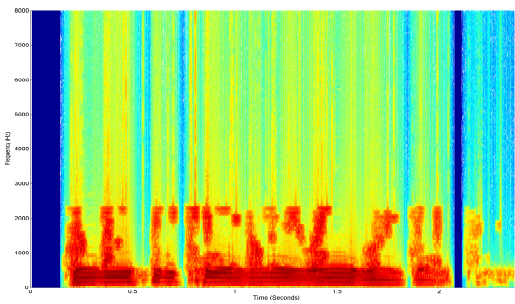
Question 10		
Coding Technique:	Original	AN version of Onset Coding
Sound Type:	Male Speech (Speech Type)	
URL:	www.cs.stir.ac.uk/~mpa/testing/questions_10.html	
Spectrogram:	 <p>Spectrogram of the original sound</p>	 <p>Here the spectrogram follows the same pattern but shows that it has been overly distributed.</p>
Test Answer (in favor):	21	0
Observed Probability:	1	0

TABLE 5.10: Test Question 10. Here all of the participants have favored the original sound over the AN-Onset-Decoded sound.

5.5.11 Question 11: Koickal's Reconstructed Sound vs AN-Decoded Sound

Here, the Temple Bell 'percussion' type sound has been played. First the Koickal's Reconstructed sound has been played followed by the AN-Decoded sound next. Table 5.11 compares these two sounds in detail.

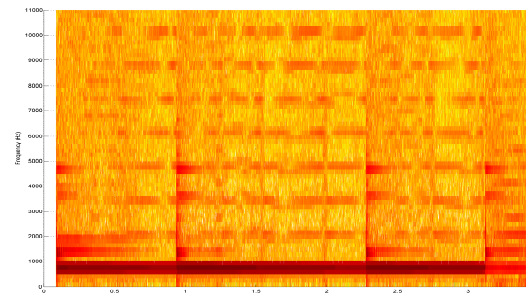
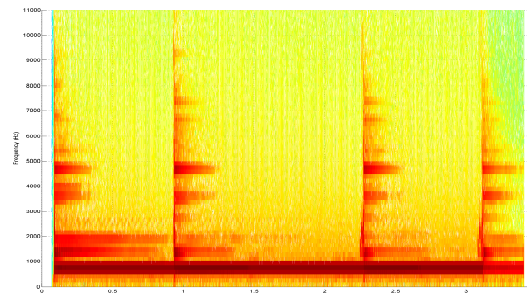
Question 11		
Coding Technique:	Koickal's Spike Coding	AN Coding
Sound Type:	Temple Bell (Percussion Type)	
URL:	www.cs.stir.ac.uk/~mpa/testing/questions_11.html	
Spectrogram:	 <p>The spectrogram of the decoded sound shows that the energy scattered almost everywhere and no sign of any pattern</p>	 <p>Here the spectrogram follows the same kind of pattern of the original one but also that it has been overly distributed.</p>
Test Answer (in favor):	0	21
Observed Probability:	0	1

TABLE 5.11: Test Question 11. Here none of the participants have favored the Koickal's Reconstructed sound over the AN-Decoded sound.

5.5.12 Question 12: AN version of Onset-Decoded Sound vs AN-Decoded Sound

Here, the Electro Stereo Guitar frequency level 4 ‘string’ type sound has been played. First the AN-Onset-Decoded sound has been played followed by the AN-Decoded sound next. Table 5.12 compares these two sounds in detail.

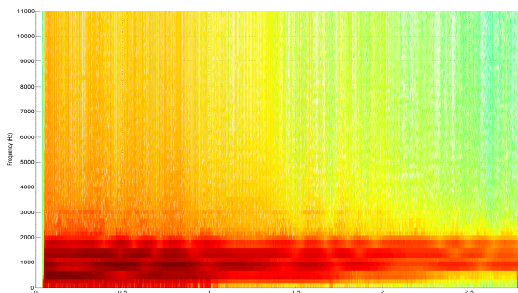
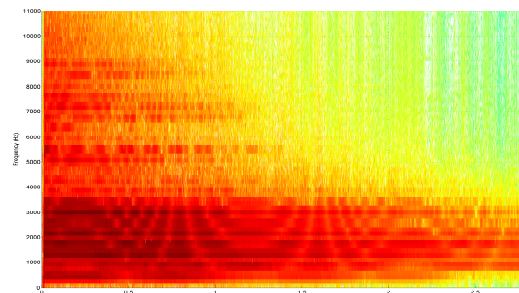
Question 12		
Coding Technique:	AN version of Onset Coding	AN Coding
Sound Type:	Electro Guitar Sound (String Type)	
URL:	www.cs.stir.ac.uk/~mpa/testing/questions_12.html	
Spectrogram:	 <p>The spectrogram of the AN-Onset-Decoded sound shows that the energy has been lost for the higher frequencies and they appear only for lower frequencies</p>	 <p>Here the spectrogram follows the same kind of pattern of the original one but also that it has been overly distributed.</p>
Test Answer (in favor):	7	14
Observed Probability:	0.3333	0.6667

TABLE 5.12: Test Question 12. Here 7 of the participants have favored the AN-Onset-Decoded sound over the AN-Decoded sound.

5.5.13 Question 13: AN-Decoded Sound vs AN version of Onset-Decoded Sound

Here, the Female ‘speech’ type of sound has been played. The AN-Decoded sound has been played first followed by AN-Onset-Decoded sound next. Table 5.13 compares these two sounds in detail.

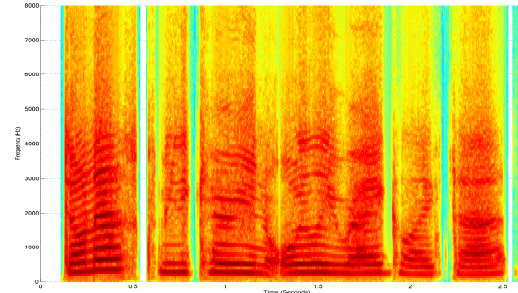
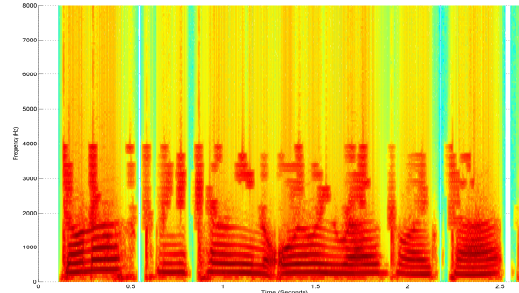
Question 13		
Coding Technique:	AN Coding	AN version of Onset Coding
Sound Type:	Female Speech (Speech Type)	
URL:	www.cs.stir.ac.uk/~mpa/testing/questions_13.html	
Spectrogram:	 <p>The spectrogram of the AN-Decoded sound shows that the energy has been scattered almost everywhere</p>	 <p>Here the spectrogram shows that the energy is quite well distributed by following the same pattern in the original sound file. However the energy is not very well distributed over 4 kHz</p>
Test Answer (in favor):	9	12
Observed Probability:	0.4286	0.5714

TABLE 5.13: Test Question 13. Here 9 of the participants have favored the AN-Decoded sound over the AN-Onset-Decoded sound.

5.5.14 Question 14: AN version of Onset-Decoded Sound vs AN-Decoded Sound

Here, the celesta ‘string’ type frequency 4 level sound has been decoded from the AN version of Onset spike code and then played and compared with the AN-Decoded sound. The first sound played is the An-Onset-Decoded sound and the second one is the decoded from AN spike codes. Table 5.14 compares these two sounds in detail.

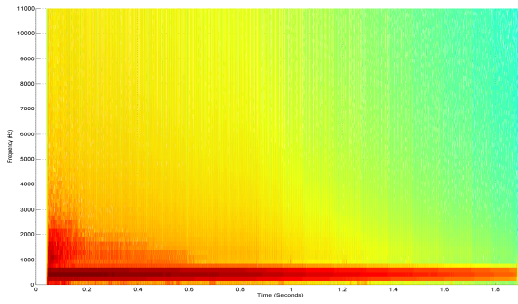
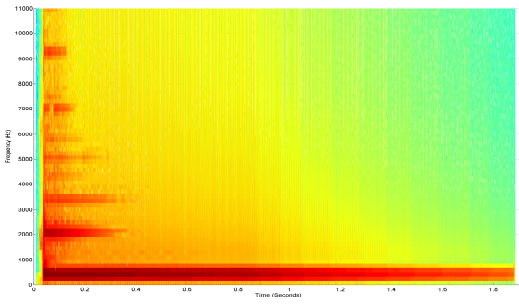
Question 14		
Coding Technique:	AN version of Onset Coding	AN Coding
Sound Type:	Celesta frequency level 4 (String Type)	
URL:	www.cs.stir.ac.uk/~mpa/testing/questions_14.html	
Spectrogram:	 <p>The spectrogram of the AN-Onset-Decoded sound shows that the energy has been lost for the higher frequencies and they appear only for lower frequencies</p>	 <p>Here the spectrogram follows the same kind of pattern of the original one but also that it has been overly distributed a little bit.</p>
Test Answer (in favor):	3	18
Observed Probability:	0.1429	0.8571

TABLE 5.14: Test Question 14. Here 3 of the participants have favored the AN-Onset-Decoded sound over the AN-Decoded sound.

5.5.15 Question 15: AN-Decoded Sound vs Original

Here, the Female ‘Speech’ type of sound has been played. The AN spike decoded sound has been played first followed by the original sound next. Table 5.15 compares these two sounds in detail.

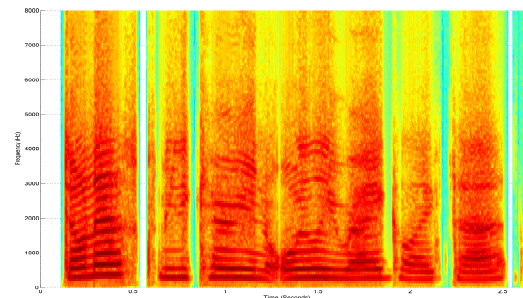
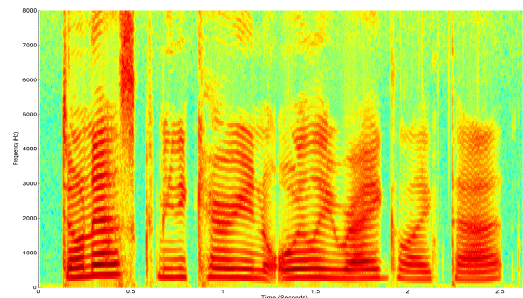
Question 15		
Coding Technique:	AN Coding	Original
Sound Type:	Female Speech (Speech Type)	
URL:	www.cs.stir.ac.uk/~mpa/testing/questions_15.html	
Spectrogram:	 <p>The spectrogram of AN-Decoded sound which shows that the energy is scattered everywhere and there are very little similarity with the original sound’s spectrogram. The energy has been distributed almost everywhere</p>	 <p>Spectrogram of the original signal.</p>
Test Answer (in favor):	0	21
Observed Probability:	0	1

TABLE 5.15: Test Question 15. Here nobody has favored the AN-decoded sound over the original sound.

5.5.16 Question 16: Original Sound vs AN-Decoded Sound

Here, the Electro Stereo Guitar frequency level 4 ‘string’ type sound has been played. First the original sound has been played followed by the AN-Decoded sound next. Table 5.16 compares these two sounds in detail.

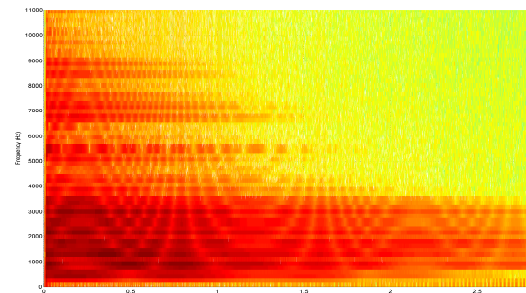
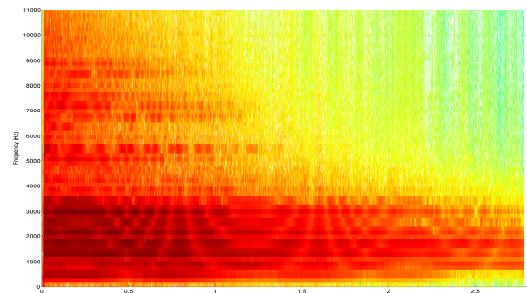
Question 16		
Coding Technique:	Original	AN Coding
Sound Type:	Electro Guitar Sound (String Type)	
URL:	www.cs.stir.ac.uk/~mpa/testing/questions_16.html	
Spectrogram:	 <p>Spectrogram of the original sound</p>	 <p>The spectrogram shows that the energy is more scattered in the decoded signal.</p>
Test Answer (in favor):	19	2
Observed Probability:	0.9048	0.0952

TABLE 5.16: Test Question 16. Here 19 out of 21 have favored the original sound over the AN-decoded sound.

5.5.17 Question 17: Koickal's Reconstructed Sound vs Original Sound

Here, the Male 'Speech' type sound has been played. First Koickal's Reconstructed sound has been played followed by the original sound next. Table 5.17 compares these two sounds in detail.

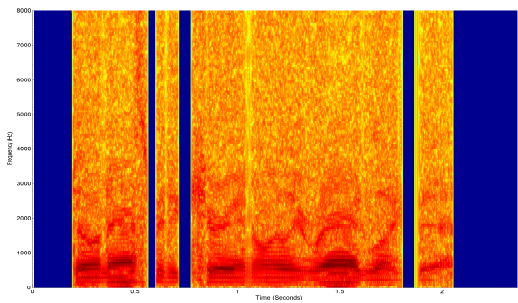
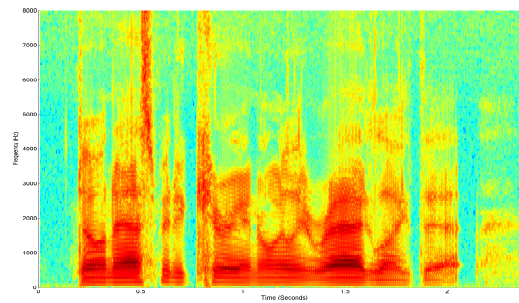
Question 17		
Coding Technique:	Koickal's Spike Coding	Original
Sound Type:	Male Speech (Speech Type)	
URL:	www.cs.stir.ac.uk/~mpa/testing/questions_17.html	
Spectrogram:	 <p>The spectrogram shows that the energy has been concentrated only in few areas and it appears only for lower frequencies.</p>	 <p>Spectrogram of the original sound.</p>
Test Answer (in favor):	0	21
Observed Probability:	0	1

TABLE 5.17: Test Question 17. Here none of the participants have favored the Koickal's Reconstructed sound over the original sound.

5.5.18 Question 18: Original Sound vs Koickal's Reconstructed Sound

Here, the Electro Stereo Guitar frequency level 4 ‘string’ type sound has been played. First the original sound has been played followed by the Koickal's Reconstructed sound next. Table 5.18 compares these two sounds in detail.

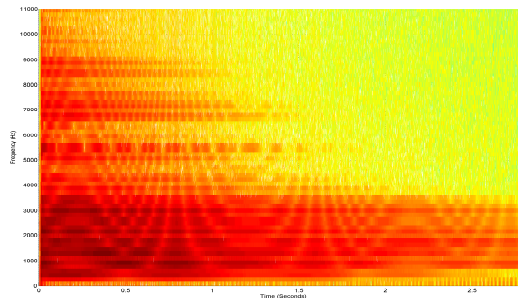
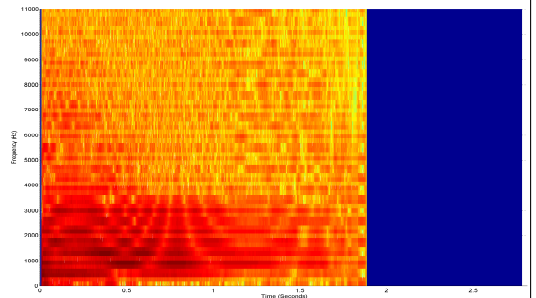
Question 18		
Coding Technique:	Original	Koickal's Spike Coding
Sound Type:	Electro Guitar Sound (String Type)	
URL:	www.cs.stir.ac.uk/~mpa/testing/questions_18.html	
Spectrogram:	 <p>Spectrogram of the original sound.</p>	 <p>The spectrogram of the Koickal's Reconstructed sound. The energy has not followed the right pattern in the original sound.</p>
Test Answer (in favor):	21	0
Observed Probability:	1	0

TABLE 5.18: Test Question 18. Here all of the participants have favored the original sound over the Koickal's Reconstructed sound.

5.5.19 Question 19: AN-Decoded Sound vs AN version of Onset-Decoded Sound

Here, the Male ‘speech’ type of sound has been played. The AN-Decoded sound has been played first followed by AN-Onset-Decoded sound next. Table 5.19 compares these two sounds in detail.

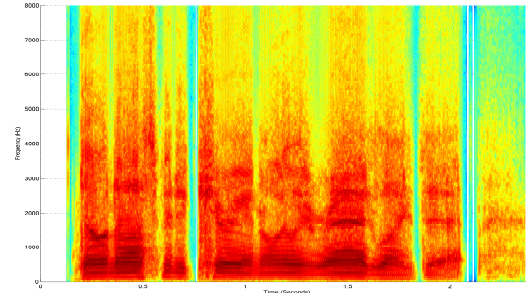
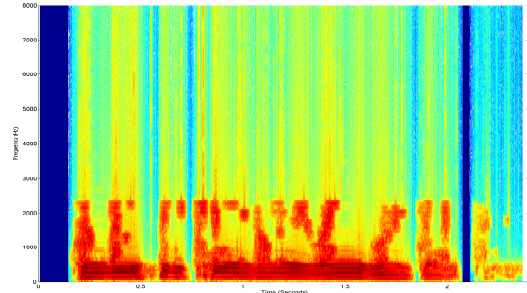
Question 19		
Coding Technique:	AN Coding	AN version of Onset Coding
Sound Type:	Male Speech (Speech Type)	
URL:	www.cs.stir.ac.uk/~mpa/testing/questions_19.html	
Spectrogram:	 <p>The spectrogram of the AN-Decoded sound shows that the energy has been scattered almost everywhere</p>	 <p>Here the spectrogram shows that the energy is quite well distributed by following the same pattern in the original sound file. However the energy is not very well distributed over 4 kHz</p>
Test Answer (in favor):	14	7
Observed Probability:	0.6667	0.3333

TABLE 5.19: Test Question 19. Here 14 of the participants have favored the AN-Decoded sound over the AN-Onset-Decoded sound.

5.5.20 Question 20: Koickal's Reconstructed Sound vs Original Sound

Here, the Female 'Speech' type sound has been played. First Koickal's Reconstructed sound has been played followed by the original sound next. Table 5.20 compares these two sounds in detail.

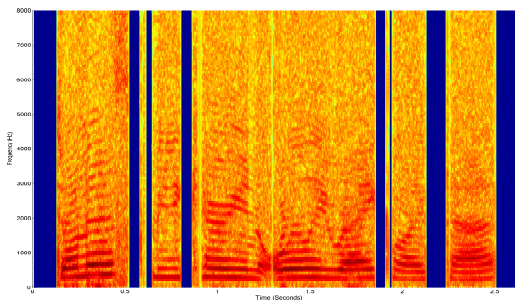
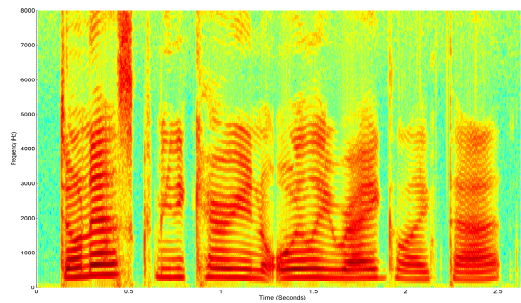
Question 20		
Coding Technique:	Koickal's Spike Coding	Original
Sound Type:	Female Speech (Speech Type)	
URL:	www.cs.stir.ac.uk/~mpa/testing/questions_20.html	
Spectrogram:	 <p>The spectrogram shows that the energy has been concentrated only in few areas and it appears only for lower frequencies.</p>	 <p>Spectrogram of the original sound.</p>
Test Answer (in favor):	0	21
Observed Probability:	0	1

TABLE 5.20: Test Question 20. Here none of the participants have favored the Koickal's Reconstructed sound over the original sound.

5.6 The Statistical Methods used to explain the answers from the Sound Test

We have used the **Binomial Distribution** to find out the better technique over the other in each question with a statistical significance level. Binomial Distribution is one of the most widely used distribution in research ([84]). The way Binomial test works has been described below briefly.

- A Binomial Experiment consists of n identical trials, each of which results in one of two outcomes, a ‘success’ or ‘failure’.
- The probabilities of success or failure is constant across trials.
- All of the trials are independent i.e. they are not affected by any other outcomes.
- The equation of Binomial test is $P(Y = y) = \frac{n!}{y!(n-y)!}p^yq^{(n-y)}$, where $P(Y = y)$ can also be determined by the Binomial Table.
- It has mean, $\mu = np$ and variance, $\sigma^2 = npq$.

The Hypothesis test can be either one tailed or two tailed. In this subjective testing, we have used two tailed hypothesis test as we are interested to find out if the responses for each technique are equally likely or not. So, a two tailed test is appropriate here. The procedure of Two-Tailed Binomial Test is

- An one-tailed Probability Value (Binomial P) should be calculated and should be doubled (Two Tailed Test P).
- If that value is less than the α , which is the probability of falsely rejecting a true null hypothesis; we can reject the null hypothesis.

We have followed this procedure in the analysis of testing. First, we have created the Null and Alternative Hypothesis and then calculated the Binomial Probability and multiplied it by 2. Then the accepted probability has been compared with 1%, 5% and 10% level of significance. After that the final conclusion about two techniques has been made.

5.7 The Explanation of answers from each questions and Evidences

In this section, we will find statistical evidences received for one particular coding technique over the other. The binomial distribution has been used to find the evidences. The structure of the binomial test has been the same, however the results are different for all the testing questions. The null hypothesis has been:

$$H_0 : \theta = 0.5 \quad (5.1)$$

And the alternative hypothesis:

$$H_1 : \theta \neq 0.5 \quad (5.2)$$

The Binomial mass function has been

$$b(x; n, p) = \binom{n}{x} p^x (1 - p)^{n-x} \quad \text{where } \binom{n}{x} = \frac{n!}{x!(n-x)!} \quad (5.3)$$

In this equation 5.3; two fixed parameters: n = number of trials and p is the probability. x is the number of successes. The binomial probabilities have been calculated by following equation 5.3 for each question below. ‘Binomial P’ represents the one-tailed probability value and ‘Two Tailed Test P’ is the double of that one-tailed probability value.

5.7.1 Question 1: Original Sound vs AN-Decoded Sound

In this question, 11 out of 21 participants favored the original over AN-Decoded sound. So, the statistics table follows in table 5.21.

	Original Sound	AN-Decoded Sound
Test Answer (in favor):	11	10
Binomial P:	0.668	0.500
Two Tailed Test P:	1.336	1.000
Accepted Probability:	1.000	
Reject H_0 at 90% C.I.:	NO	
Reject H_0 at 95% C.I.:	NO	
Reject H_0 at 99% C.I.:	NO	
Conclusion:	At, 90%, 95% and 99% confidence level, we do not have enough statistical evidence to say: ‘The Original sound and AN-Decoded sound is different from each other.’	

TABLE 5.21: Test Question 1. The AN-Decoded sound is not different from the Original.

5.7.2 Question 2: Original Sound vs AN Version of Onset-Decoded Sound

In this question, 19 out of 21 participants favored the original over AN-Onset-Decoded sound. So, the statistics table follows in table 5.22.

	Original Sound	AN-Onset-Decoded Sound
Test Answer (in favor):	19	2
Binomial P:	1.000	0.000
Two Tailed Test P:	2.000	0.000
Accepted Probability:	0.000	
Reject H_0 at 90% C.I.:	YES	
Reject H_0 at 95% C.I.:	YES	
Reject H_0 at 99% C.I.:	YES	
Conclusion:	At, 90%, 95% and 99% confidence level, there is enough statistical evidence to say: ‘The Original sound and AN-Onset-Decoded sound is different from each other.’	

TABLE 5.22: Test Question 2. The AN-Onset-Decoded sound is different from the Original. The statistical evidences also suggest that original sound is better than the AN-Onset-Coded sound.

5.7.3 Question 3: AN Version of Onset-Decoded Sound vs Original

In this question, only 1 out of 21 participants favored the AN-Onset-Decoded sound over Original sound. So, the statistics table follows in table 5.23.

	AN-Onset-Decoded Sound	Original Sound
Test Answer (in favor):	1	20
Binomial P:	0.000	1.000
Two Tailed Test P:	0.000	2.000
Accepted Probability:	0.000	
Reject H_0 at 90% C.I.:	YES	
Reject H_0 at 95% C.I.:	YES	
Reject H_0 at 99% C.I.:	YES	
Conclusion:	At, 90%, 95% and 99% confidence level, there is enough statistical evidence to say: 'The AN-Onset-Decoded sound and Original sound is different from each other.'	

TABLE 5.23: Test Question 3. The Original sound is different from the AN-Onset-Decoded sound. The statistical evidences also suggest that AN-Onset-Decoded sound is worse than the Original sound.

5.7.4 Question 4: AN-Decoded Sound vs Original

In this question, nobody among 21 participants favored the AN-Decoded sound over Original sound. The statistics table follows in table 5.24.

	AN-Decoded Sound	Original Sound
Test Answer (in favor):	0	21
Binomial P:	0.000	1.000
Two Tailed Test P:	0.000	2.000
Accepted Probability:	0.000	
Reject H_0 at 90% C.I.:	YES	
Reject H_0 at 95% C.I.:	YES	
Reject H_0 at 99% C.I.:	YES	
Conclusion:	At, 90%, 95% and 99% confidence level, there is enough statistical evidence to say: ‘The AN-Decoded sound and Original sound is different from each other.’	

TABLE 5.24: Test Question 4. The Original sound is different from the AN-Decoded sound. The statistical evidences also suggest that AN-Decoded sound is worse than the Original sound.

5.7.5 Question 5: AN-Decoded Sound vs AN version of Onset-Decoded Sound

In this question, 19 out of 21 participants favored the AN-Decoded sound over AN-Onset-Decoded sound. The statistics table follows in table 5.25.

	AN-Decoded Sound	AN-Onset-Decoded Sound
Test Answer (in favor):	19	2
Binomial P:	1.000	0.000
Two Tailed Test P:	2.000	0.000
Accepted Probability:	0.000	
Reject H_0 at 90% C.I.:	YES	
Reject H_0 at 95% C.I.:	YES	
Reject H_0 at 99% C.I.:	YES	
Conclusion:	At, 90%, 95% and 99% confidence level, there is enough statistical evidence to say: ‘The AN-Decoded sound and AN-Onset-Decoded sound is different from each other.’	

TABLE 5.25: Test Question 5. The AN-Decoded sound is different from the AN-Onset-Decoded sound. The statistical evidences also suggest that AN-Decoded sound is better than the AN-Onset-Decoded sound.

5.7.6 Question 6: Original vs AN version of Onset-Decoded Sound

In this question, all of the participants favored the original over AN-Onset-Decoded sound. So, the statistics table follows in table 5.26.

	Original Sound	AN-Onset-Decoded Sound
Test Answer (in favor):	21	0
Binomial P:	1.000	0.000
Two Tailed Test P:	2.000	0.000
Accepted Probability:	0.000	
Reject H_0 at 90% C.I.:	YES	
Reject H_0 at 95% C.I.:	YES	
Reject H_0 at 99% C.I.:	YES	
Conclusion:	At, 90%, 95% and 99% confidence level, there is enough statistical evidence to say: 'The Original sound and AN-Onset-Decoded sound is different from each other.'	

TABLE 5.26: Test Question 6. The AN-Onset-Decoded sound is different from the Original. The statistical evidences also suggest that original sound is better than the AN-Onset-Decoded sound.

5.7.7 Question 7: AN-Decoded Sound vs Koickal's Reconstructed Sound

In this question, all of the participants favored the AN-Decoded sound over Koickal-Reconstructed sound. The statistics table follows in table 5.27.

	AN-Decoded Sound	Koickal-Reconstructed Sound
Test Answer (in favor):	21	0
Binomial P:	1.000	0.000
Two Tailed Test P:	2.000	0.000
Accepted Probability:	0.000	
Reject H_0 at 90% C.I.:	YES	
Reject H_0 at 95% C.I.:	YES	
Reject H_0 at 99% C.I.:	YES	
Conclusion:	At, 90%, 95% and 99% confidence level, there is enough statistical evidence to say: 'The AN-Decoded sound and Koickal-Reconstructed sound is different from each other.'	

TABLE 5.27: Test Question 7. The AN-Decoded sound is different from the Koickal-Reconstructed sound. The statistical evidences also suggest that AN-Decoded sound is better than the Koickal-Reconstructed sound.

5.7.8 Question 8: Original vs Koickal's Reconstructed Sound

In this question, all of the participants favored the Original sound over Koickal-Reconstructed sound. The statistics table follows in table 5.28.

	Original Sound	Koickal-Reconstructed Sound
Test Answer (in favor):	21	0
Binomial P:	1.000	0.000
Two Tailed Test P:	2.000	0.000
Accepted Probability:	0.000	
Reject H_0 at 90% C.I.:	YES	
Reject H_0 at 95% C.I.:	YES	
Reject H_0 at 99% C.I.:	YES	
Conclusion:	At, 90%, 95% and 99% confidence level, there is enough statistical evidence to say: 'The Original sound and Koickal-Reconstructed sound is different from each other.'	

TABLE 5.28: Test Question 8. The Original sound is different from the Koickal-Reconstructed sound. The statistical evidences also suggest that Original sound is better than the Koickal-Reconstructed sound.

5.7.9 Question 9: AN version of Onset-Decoded Sound vs Koickal's Reconstructed Sound

In this question, 16 out of all the participants favored the AN-Onset-Decoded sound over Koickal-Reconstructed sound. The statistics table follows in table 5.29.

	AN-Onset-Decoded Sound	Koickal-Reconstructed Sound
Test Answer (in favor):	16	5
Binomial P:	0.996	0.013
Two Tailed Test P:	1.993	0.027
Accepted Probability:	0.027	
Reject H_0 at 90% C.I.:	YES	
Reject H_0 at 95% C.I.:	YES	
Reject H_0 at 99% C.I.:	NO	
Conclusion:	At, 90% and 95% confidence level, there is enough statistical evidence to say: 'The AN-Onset-Decoded sound and Koickal-Reconstructed sound is different from each other.' However at 99% confidence level, we do NOT have enough statistical evidence to say: 'The AN-Onset-Decoded sound and Koickal-Reconstructed sound is different from each other.'	

TABLE 5.29: Test Question 9. The AN-Onset-Decoded sound is different from the Koickal-Reconstructed sound in 90 or 95 times out of 100. The statistical evidences also suggest that AN-Onset-Decoded sound is better than the Koickal-Reconstructed sound. However for 99 times out of 100, we do not have such evidences.

5.7.10 Question 10: Original vs AN version of Onset-Decoded Sound

In this question, all of the participants favored the Original sound over AN-Onset-Decoded sound. The statistics table follows in table 5.30.

	Original Sound	AN-Onset-Decoded Sound
Test Answer (in favor):	21	0
Binomial P:	1.000	0.000
Two Tailed Test P:	2.000	0.000
Accepted Probability:	0.000	
Reject H_0 at 90% C.I.:	YES	
Reject H_0 at 95% C.I.:	YES	
Reject H_0 at 99% C.I.:	YES	
Conclusion:	At, 90%, 95% and 99% confidence level, there is enough statistical evidence to say: 'The Original sound and AN-Onset-Decoded sound is different from each other.'	

TABLE 5.30: Test Question 10. The Original sound is different from the AN-Onset-Decoded sound. The statistical evidences also suggest that Original sound is better than the AN-Onset-Decoded sound.

5.7.11 Question 11: Koickal's Reconstructed Sound vs AN-Decoded Sound

In this question, all of the participants favored the Koickal-Reconstructed sound over AN-Decoded sound. The statistics table follows in table 5.31.

	Koickal-Reconstructed Sound	AN-Decoded Sound
Test Answer (in favor):	0	21
Binomial P:	0.000	1.000
Two Tailed Test P:	0.000	2.000
Accepted Probability:	0.000	
Reject H_0 at 90% C.I.:	YES	
Reject H_0 at 95% C.I.:	YES	
Reject H_0 at 99% C.I.:	YES	
Conclusion:	At, 90%, 95% and 99% confidence level, there is enough statistical evidence to say: 'The Koickal-Reconstructed sound and AN-Decoded sound is different from each other.'	

TABLE 5.31: Test Question 11. The Koickal-Reconstructed sound is different from the AN-Decoded sound. The statistical evidences also suggest that Koickal-Reconstructed sound is worse than the AN-Decoded sound.

5.7.12 Question 12: AN version of Onset-Decoded Sound vs AN-Decoded Sound

In this question, all of the participants favored the AN-Onset-Decoded sound over AN-Decoded sound. The statistics table follows in table 5.32.

	AN-Onset-Decoded Sound	AN-Decoded Sound
Test Answer (in favor):	7	14
Binomial P:	0.095	0.961
Two Tailed Test P:	0.189	1.922
Accepted Probability:	0.189	
Reject H_0 at 90% C.I.:	NO	
Reject H_0 at 95% C.I.:	NO	
Reject H_0 at 99% C.I.:	NO	
Conclusion:	At, 90%, 95% and 99% confidence level, there is NOT enough statistical evidence to say: ‘The AN-Onset-Decoded sound and AN-Decoded sound is different from each other.’	

TABLE 5.32: Test Question 12. The AN-Onset-Decoded sound is NOT different from the AN-Decoded sound.

5.7.13 Question 13: AN-Decoded Sound vs AN version of Onset-Decoded Sound

In this question, all of the participants favored the AN-Decoded sound over AN-Onset-Decoded sound. The statistics table follows in table 5.33.

	AN-Decoded Sound	AN-Onset-Decoded Sound
Test Answer (in favor):	9	12
Binomial P:	0.332	0.808
Two Tailed Test P:	0.664	1.617
Accepted Probability:	0.664	
Reject H_0 at 90% C.I.:	NO	
Reject H_0 at 95% C.I.:	NO	
Reject H_0 at 99% C.I.:	NO	
Conclusion:	At, 90%, 95% and 99% confidence level, there is NOT enough statistical evidence to say: ‘The AN-Decoded sound and AN-Onset-Decoded sound is different from each other.’	

TABLE 5.33: Test Question 12. The AN-Decoded sound is NOT different from the AN-Onset-Decoded sound.

5.7.14 Question 14: AN version of Onset-Decoded Sound vs AN-Decoded Sound

In this question, all of the participants favored the AN-Onset-Decoded sound over AN-Decoded sound. The statistics table follows in table 5.34.

	AN-Onset-Decoded Sound	AN-Decoded Sound
Test Answer (in favor):	3	18
Binomial P:	0.001	1.000
Two Tailed Test P:	0.001	2.000
Accepted Probability:	0.001	
Reject H_0 at 90% C.I.:	YES	
Reject H_0 at 95% C.I.:	YES	
Reject H_0 at 99% C.I.:	YES	
Conclusion:	At, 90%, 95% and 99% confidence level, there is enough statistical evidence to say: ‘The AN-Onset-Decoded sound and AN-Decoded sound is different from each other.’	

TABLE 5.34: Test Question 14. The AN-Onset-Decoded sound is different from the AN-Decoded sound. The statistical evidences also suggest that AN-Onset-Decoded sound is worse than the AN-Decoded sound.

5.7.15 Question 15: AN-Decoded Sound vs Original

In this question, all of the participants favored the AN-Decoded sound over Original sound. The statistics table follows in table 5.35.

	AN-Decoded Sound	Original Sound
Test Answer (in favor):	0	21
Binomial P:	0.000	1.000
Two Tailed Test P:	0.000	2.000
Accepted Probability:	0.000	
Reject H_0 at 90% C.I.:	YES	
Reject H_0 at 95% C.I.:	YES	
Reject H_0 at 99% C.I.:	YES	
Conclusion:	At, 90%, 95% and 99% confidence level, there is enough statistical evidence to say: 'The AN-Decoded sound and Original sound is different from each other.'	

TABLE 5.35: Test Question 15. The AN-Decoded sound is different from the Original sound. The statistical evidences also suggest that AN-Decoded sound is worse than the Original sound.

5.7.16 Question 16: Original Sound vs AN-Decoded Sound

In this question, almost everyone (19) out of the participants favored the Original sound over the AN-Decoded sound. The statistics table follows in table 5.36.

	Original Sound	AN-Decoded Sound
Test Answer (in favor):	19	2
Binomial P:	1.000	0.000
Two Tailed Test P:	2.000	0.000
Accepted Probability:	0.000	
Reject H_0 at 90% C.I.:	YES	
Reject H_0 at 95% C.I.:	YES	
Reject H_0 at 99% C.I.:	YES	
Conclusion:	At, 90%, 95% and 99% confidence level, there is enough statistical evidence to say: 'The Original sound and AN-Decoded sound is different from each other.'	

TABLE 5.36: Test Question 16. The Original sound is different from the AN-Decoded sound. The statistical evidences also suggest that Original sound is better than the AN-Decoded sound.

5.7.17 Question 17: Koickal's Reconstructed Sound vs Original Sound

In this question, nobody favored the Koickal-Reconstructed sound over Original sound.

The statistics table follows in table 5.37.

	Koickal-Reconstructed Sound	Original Sound
Test Answer (in favor):	0	21
Binomial P:	0.000	1.000
Two Tailed Test P:	0.000	2.000
Accepted Probability:	0.000	
Reject H_0 at 90% C.I.:	YES	
Reject H_0 at 95% C.I.:	YES	
Reject H_0 at 99% C.I.:	YES	
Conclusion:	At, 90%, 95% and 99% confidence level, there is enough statistical evidence to say: 'The Koickal-Reconstructed sound and Original sound is different from each other.'	

TABLE 5.37: Test Question 17. The Koickal-Reconstructed sound is different from the Original sound. The statistical evidences also suggest that Koickal-Reconstructed sound is worse than the Original sound.

5.7.18 Question 18: Original Sound vs Koickal's Reconstructed Sound

In this question, everybody favored the Original sound over Koickal-Reconstructed sound. The statistics table follows in table 5.38.

	Original Sound	Koickal-Reconstructed Sound
Test Answer (in favor):	21	0
Binomial P:	1.000	0.000
Two Tailed Test P:	2.000	0.000
Accepted Probability:	0.000	
Reject H_0 at 90% C.I.:	YES	
Reject H_0 at 95% C.I.:	YES	
Reject H_0 at 99% C.I.:	YES	
Conclusion:	At, 90%, 95% and 99% confidence level, there is enough statistical evidence to say: 'The Original sound and Koickal-Reconstructed sound is different from each other.'	

TABLE 5.38: Test Question 18. The Original sound is different from the Koickal-Reconstructed sound. The statistical evidences also suggest that Original sound is better than the Koickal-Reconstructed sound.

5.7.19 Question 19: AN-Decoded Sound vs AN version of Onset-Decoded Sound

In this question, 14 of the participants favored the AN-Decoded sound over AN-Onset-Decoded sound. The statistics table follows in table 5.39.

	AN-Decoded Sound	AN-Onset-Decoded Sound
Test Answer (in favor):	14	7
Binomial P:	0.961	0.095
Two Tailed Test P:	1.922	0.189
Accepted Probability:	0.189	
Reject H_0 at 90% C.I.:	NO	
Reject H_0 at 95% C.I.:	NO	
Reject H_0 at 99% C.I.:	NO	
Conclusion:	At, 90%, 95% and 99% confidence level, there is NOT enough statistical evidence to say: ‘The AN-Decoded sound and AN-Onset-Decoded sound is different from each other.’	

TABLE 5.39: Test Question 19. The AN-Decoded sound is NOT different from the AN-Onset-Decoded sound.

5.7.20 Question 20: Koickal's Reconstructed Sound vs Original Sound

In this question, nobody favored the Koickal-Reconstructed sound over Original sound. The statistics table follows in table 5.40.

	Koickal-Reconstructed Sound	Original Sound
Test Answer (in favor):	0	21
Binomial P:	0.000	1.000
Two Tailed Test P:	0.000	2.000
Accepted Probability:	0.000	
Reject H_0 at 90% C.I.:	YES	
Reject H_0 at 95% C.I.:	YES	
Reject H_0 at 99% C.I.:	YES	
Conclusion:	At, 90%, 95% and 99% confidence level, there is enough statistical evidence to say: 'The Koickal-Reconstructed sound and Original sound is different from each other.'	

TABLE 5.40: Test Question 20. The Koickal-Reconstructed sound is different from the Original sound. The statistical evidences also suggest that Koickal-Reconstructed sound is worse than the Original sound.

So, all of the questions have been detailed by the corresponding statistical evidences behind each decision been made.

5.8 Sound Testing Conclusion and Remarks

The results obtained from the subjective testing produce this table:

Questions	Sound Type	Techniques Used		Better Technique
Question 1	String	Original	AN-Decoded	SAME
Question 2	String	Original	AN-Onset-Decoded	BETTER
Question 3	Percussion	AN-Onset-Decoded	Original	WORSE
Question 4	Male Speech	AN-Decoded	Original	WORSE
Question 5	Percussion	AN-Decoded	AN-Onset-Decoded	BETTER
Question 6	Female Speech	Original	AN-Onset-Decoded	BETTER
Question 7	String	AN-Decoded	Koickal-Decoded	BETTER
Question 8	String	Original	Koickal-Decoded	BETTER
Question 9	String	AN-Onset-Decoded	Koickal-Decoded	Better/Same
Question 10	Male Speech	Original	AN-Onset-Decoded	BETTER
Question 11	Percussion	Koickal-Decoded	AN-Decoded	WORSE
Question 12	String	AN-Onset-Decoded	AN-Decoded	SAME
Question 13	Female Speech	AN-Decoded	AN-Onset-Decoded	SAME
Question 14	String	AN-Onset-Decoded	AN-Decoded	WORSE
Question 15	Female Speech	AN-Decoded	Original	WORSE
Question 16	String	Original	AN-Decoded	BETTER
Question 17	Male Speech	Koickal-Decoded	Original	WORSE
Question 18	String	Original	Koickal-Decoded	BETTER
Question 19	Male Speech	AN-Decoded	AN-Onset-Decoded	SAME
Question 20	Female Speech	Koickal-Decoded	Original	WORSE

TABLE 5.41: Test Summary: Here the test results have been summed up. This table shows the techniques which are better than or same as the others for each ‘string’, ‘percussion’, ‘male voice’ and ‘female voice’.

Table 5.41 sums up the entire subjective testing and the possible outcomes and conclusions.

For **String type** of sound we can say

- In the table 5.41, question 1 suggests that original sound's quality is the same as the quality of the AN-Decoded sound, however question 16 suggests that original sound's quality is better than AN-Decoded sound. So, it has been concluded that Original Sound's quality is almost equal to the AN-Decoded sound's quality.
- According to table 5.41, AN-Decoded sound is as good as AN-Onset-Decoded sound as well.
- AN-Onset-Decoded Sound is not much different from Koickal-Reconstructed sound either.

So, this spike (or event) based representation works very well with 'string' type of sounds. From figure 5.1 and figure 5.2, it can be seen that this 'string' type of sound has most of its energy concentrated at certain (specially low) frequency (figure 5.1 and figure 5.2), unlike other speech or voice sound. As for low frequencies, the spike coding is less lossy than higher frequencies. That's why this spike (event) based coding technique works well for them.

For **Percussion type** of sound, we can say

- Original Sound's quality is better than the AN-Decoded sound's quality.
- AN-Decoded sound is better than AN-Onset-Decoded sound as well.
- AN-Onset-Decoded Sound is better than Koickal-Reconstructed sound.

We can say from this result that the spike or event based coding technique is definitely not good to represent this 'percussion' type of sounds. The spectrogram of this type of sounds (see figure 5.4) shows that the energy is concentrated on certain frequency bands but it is well distributed along with the time line.

For **Male Voice type** of sound, we can say

- Original Sound's quality is better than the AN-Decoded sound's quality.

- AN-Decoded sound's quality is equal to the AN-Onset-Decoded sound as well. Or, the participants were not able to distinguish any difference between the qualities of both of them.
- AN-Onset-Decoded Sound is better than Koickal-Reconstructed sound.

So, for the Male voice type of sounds, the AN spike coding works quite well. There is a distinction between the original sound and AN-Decoded sound as the lossy coding technique like spike cannot compete with the original sound.

For **Female Voice type** of sound, we can say

- Original Sound's quality is better than the AN-Decoded sound's quality.
- AN-Decoded sound's quality is equal to the AN-Onset-Decoded sound as well. Or, the participants were not able to distinguish any difference between the qualities of both of them.
- AN-Onset-Decoded Sound is better than Koickal-Reconstructed sound.

This is same as the male voice. The AN and AN-Onset spike representations are really good for the voice type of sounds for both male and female.

5.9 Discussion of Sound Testing Conclusion in Light of Number of Spikes

The number of spikes generated for each type of spike coding technique is crucial to evaluate the quality of decoded sound. If a reasonable good quality of sound can be decoded by using much fewer spikes, we can conclude that coding technique is more efficient. In the table 5.42, three different types of sounds mentioned in section 5.4.1 has been evaluated by the number of spikes generated for each coding type.

Number of Spikes for Different types of Sounds and their Lengths					
Sound Type	Sound	Sound Length	AN Spikes per second	AN_Onset Spikes per second	Koickal Spikes per second
String	Celesta F4	1.8902 sec	14635.5	4598.5	37317.7
	Celesta F7	0.98875 sec	56858.7	2383.8	43693.6
	Electric Guitar	2.7842 sec	71480.5	13939.4	45905.1
Percussion	Temple Bell	3.4115 sec	33438.4	13201.5	51089.8
Male Speech	Speech 1	3.0976 sec	49787.6	3514.3	13174.7
	Speech 2	2.3744 sec	47156.8	3472.9	10809.0
	Speech 3	2.8288 sec	49647.6	3628.4	13168.5
	Speech 4	2.336 sec	48468.3	3357.4	9965.8
	Speech 5	3.2513 sec	48421.2	3475.2	13793.2
Female Speech	Speech 1	2.8288 sec	50055.5	9148.8	14323.0
	Speech 2	3.5008 sec	51670.2	9141.3	14165.6
	Speech 3	3.328 sec	50391.8	8743.4	14085.3
	Speech 4	2.5473 sec	54499.3	9546.2	13304.3
	Speech 5	2.6304 sec	48467.9	8781.2	12977.9
Average	———	———	48212.3	6923.3	21983.4

TABLE 5.42: **Number of spikes:** Here the number of spikes per second for four different types of sounds have been mentioned along with the corresponding length in seconds. Koickal's spikes are 54.4% lower in number and AN_Onset spikes are 85.6% lower in number than AN spikes. AN_Onset spikes are 68.5% lower in number than Koickal's spikes.

In this table 5.42, it is seen that AN and Koickal's coding technique always produces many more spikes than AN_Onset spike coding technique. For example, Celesta F4 sound produces 27664 AN spikes and 70538 Koickal's spikes whereas Celesta F7 produces 56219 AN spikes and 43202 Koickal's spikes. Celesta F4 is 1.9 seconds long and Celesta F7 is 1 second long. AN spike coding technique produces spikes based on the frequency levels. Celesta F7 has more high frequency contents than Celesta F4 as the number of spikes increases with the increase of frequency. However Koickal's spike coding technique does not reflect this. However for Koickal's coding technique produces fewer spikes for speech. A 3.4 seconds long temple bell sound produces 174293 Koickal's spikes but 3.1 seconds long male speech generates only 40810 Koickal's spikes. Figure 5.7 reflects this fact.

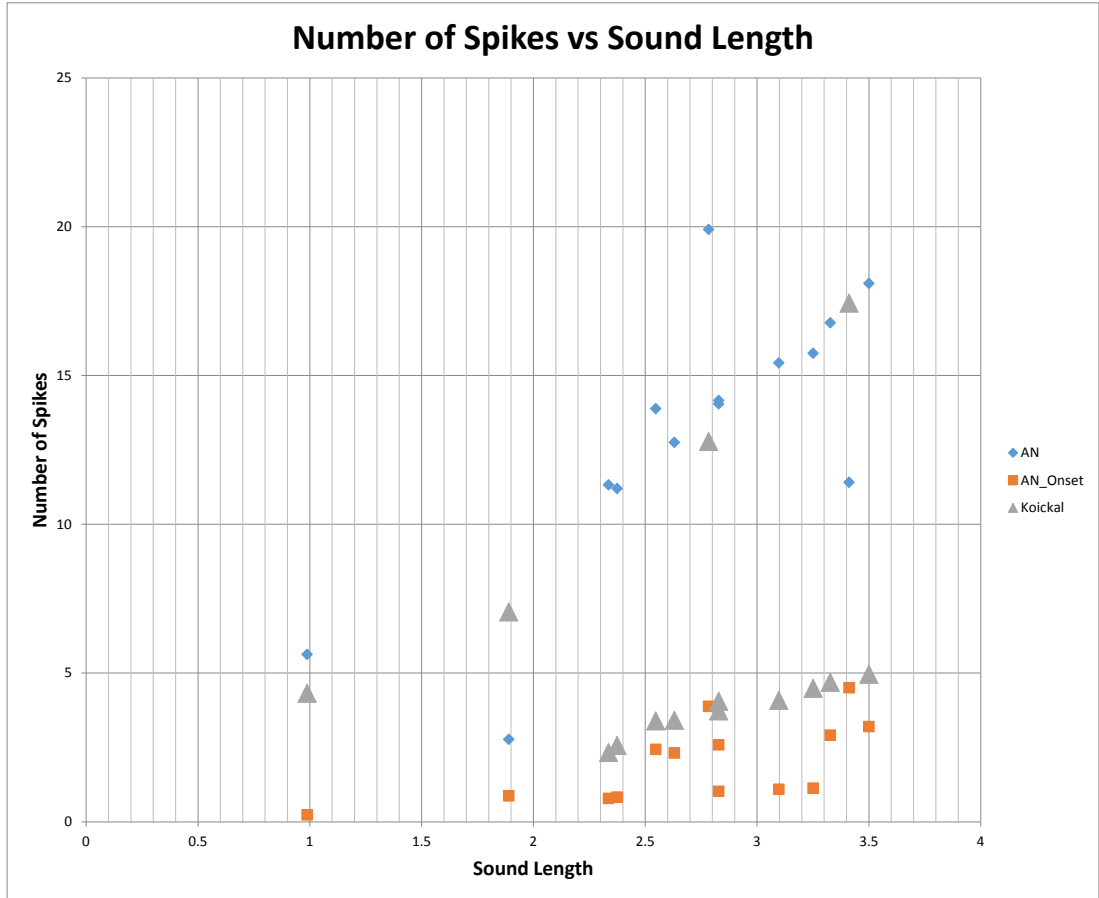


FIGURE 5.7: **Number of spikes for various sound lengths in seconds:** The number of spikes have been divided by 10000 to scale it along with lengths in seconds.

According to figure 5.7, AN spikes and Koickal's spikes follow the length of the sound with some exceptions. Figure 5.8 shows that for Celesta sounds, the number of Koickal's spikes and AN_Onset spikes decreased for shorter sound length, but AN spike's number increased. For male and female speech, the number of AN spikes had a standard deviation 21200.55 i.e. the number deviates a great deal from one another. But for AN_Onset and Koickal's spike have a standard deviation of 8944.3 and 8212.5 (rather low).

Previously in section 5.8 we have concluded that AN spikes are better representation of sounds as it can decode good quality of sounds. In this section, we have seen that the number of AN spikes are much higher than others as well. This explains that AN spike coding technique is less lossy coding technique which is able to code sound better than two other coding techniques. We have also concluded at section 5.8 that AN_Onset spikes are sometimes better but no worse than Koickal's coding technique. In this section we can also find out that the number of spikes for Koickal's coding technique is much

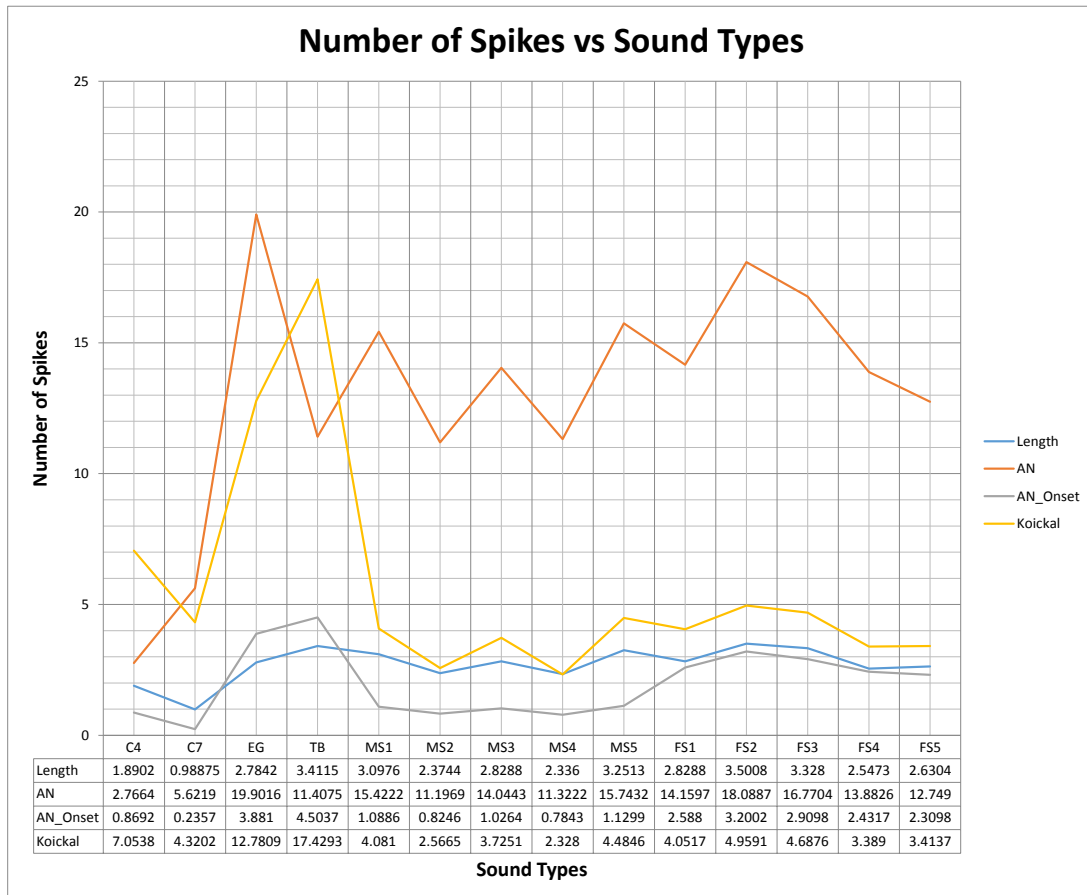


FIGURE 5.8: **Number of spikes for different sound types:** The number of spikes have been divided by 10000 to scale it along with lengths in seconds. The sound length has been measured in seconds.

higher than the AN_Onset coding technique which generally outperforms the Koickal's coding technique.

5.10 Possible Future Sound Testing and Pros and Cons

This testing produces some good evidences based on statistics about which spike or event based code is more useful to code a sound file. However there can be some future testing to be commenced so that the knowledge about these coding becomes more accurate. These are some ideas which can be followed for any future testing based on these spike or event based coding techniques:

- There are only 21 participants who participated in this testing. 15 out them are male and 6 are female. So, more female participants can be involved in this testing.
- Most of them don't have any hearing problems except only one person. He had problems hearing high frequency sounds. More participants can be involved who got some kind of hearing problems so that we can find out which spike coding technique is more useful for the hearing impaired people.
- The age group is well organized. The youngest person is 21 years old and the oldest person is 61. As the young people are more likely to hear the high frequency sounds, more young people can be hired for testing. This might have a very significant effect on the results of testing as the spike coding generates different sound for the higher frequencies. The spike generation technique has been specially tuned for middle and low frequencies as human ear is not very sensitive to the higher frequency like 10kHz or higher.
- The Male and Female speech sounds are taken from the 'Timit' dataset [72] and they are low-passed filtered. So, any frequency contents higher than 8 kHz is not present in those sounds. Figure 5.5 and 5.6 explains that there is no energy contents higher than 8 kHz. However, in the AN-Reconstruction technique the GMF has the frequency range from 100 Hz till 10,000 Hz or 10 kHz. So, in the AN-Decoded Sound there are some extra noises. For the future testing this issue can be avoided by selecting a simple and original voice file not bandpassed filtered at all. The reason the 'Timit' dataset [72] has been used in this testing is that it is very well known and available.

Chapter 6

Objective Testing & Findings

6.1 The Purpose of Objective Testing

As the second part of the testing, an objective testing has been carried out, which provides test scores which allows us to compare the three spike coding techniques mentioned in section 5.4.1 in chapter 5. This uses a comparison sound test called ‘Perceptual Evaluation of Speech Quality’ (PESQ) which has been described at section 2.6.1 in chapter 2. This test provides test scores which are useful to compare the sound quality of the original and the decoded sound.

6.2 Test Equipments & Procedure

As this is an objective test, no volunteers were required. MATLAB files for PESQ and composite test have been obtained (mentioned in Appendix D) from ([65]) and ([66]). Then they are run in MATLAB and two sounds were compared with each other and the test score have been obtained.

Test scores are put into Microsoft Excel and plotted later in this chapter. Both test equipment and procedure are very straight forward and simple and it takes only a few seconds to run each test and obtain the corresponding test scores, unlike subjective testing.

6.3 PESQ Results for Subjective Tests

The same sounds which were used in the ‘Sound Test’, has been tested under ‘PESQ’ test and the results are detailed in table 6.1 and table 6.2. In those tables, the PESQ and composite test scores has been mentioned so that we can understand how good the decoded sound is compared to the original. The maximum possible test scores of PESQ and the composite tests are obtained by comparing two identical sounds and they are:

PESQ PESQ has maximum value 4.5000

Signal Distortion C_{sig} has maximum value 5.8065

Noise Distortion C_{bak} has maximum value 5.9900

Overall Comparison C_{ovl} has maximum value 5.2165

All the other test scores will be compared with these maximum test scores and we will comment on how short they fall from those maximum scores. In table 6.1 and table 6.2, third column has been used to restate the compared techniques in the Sound Test questions. The PESQ and the composite measures have been evaluated by comparing the better sound (clean sound) with the worse sound (enhanced sound). So, for example, in the question 2 of table 5.41, we have evidence to say that the original sound is better than the AN-Onset-Decoded Sound.

So, the MATLAB function has been used like: ‘[pesq_mos]=pesq(sfreq,original.wav,AN-Onset-Decoded.wav)’, so that the worse sound can be measured with respect to the better one. In question 12 of table 6.1 and table 6.2, AN-Decoded and the AN-Onset-Decoded sounds are same or no better than the other. So, the MATLAB function has been used like: ‘[pesq_mos]=pesq(sfreq,AN-Decoded.wav,AN-Onset-Decoded.wav)’. Here, the AN-Decoded sound has been taken as better because for the other questions it is usually the case that the AN-Onset sound is better than the AN-Onset-Decoded sound.

The ‘pesq’ function is not symmetric i.e. $pesq(f_s, A, B) \neq pesq(f_s, B, A)$. In, $pesq(f_s, A, B)$, A always stands for the original or the better sound, B stands for the decoded or worse sound.

PESQ & Composite Test Scores			
Questions	Sound Type	Techniques Compared (Better vs Worse)	PESQ & Composite Test Score
Question 1	String	Original vs AN-Decoded	PESQ = 3.2872 $C_{sig} = 4.0522$ $C_{bak} = 2.4763$ $C_{ovl} = 3.6415$
Question 2	String	Original vs AN-Onset-Decoded	PESQ = 2.5923 $C_{sig} = 1.8686$ $C_{bak} = 1.7615$ $C_{ovl} = 1.8786$
Question 3	Percussion	Original vs AN-Onset-Decoded	PESQ = 2.1075 $C_{sig} = 2.3825$ $C_{bak} = 1.904$ $C_{ovl} = 2.1103$
Question 4	Male Speech	Original vs AN-Decoded	PESQ = 3.5198 $C_{sig} = 3.5793$ $C_{bak} = 2.8594$ $C_{ovl} = 3.5705$
Question 5	Percussion	AN-Decoded vs AN-Onset-Decoded	PESQ = 2.3702 $C_{sig} = 2.4483$ $C_{bak} = 3.2808$ $C_{ovl} = 2.3405$
Question 6	Female Speech	Original vs AN-Onset-Decoded	PESQ = 2.5693 $C_{sig} = 1.3955$ $C_{bak} = 2.3177$ $C_{ovl} = 1.9657$
Question 7	String	AN-Decoded vs Koickal-Decoded	PESQ = 1.9657 $C_{sig} = 0$ $C_{bak} = 1.5077$ $C_{ovl} = 0.1585$
Question 8	String	Original vs Koickal-Decoded	PESQ = 1.2275 $C_{sig} = 0$ $C_{bak} = 1.9935$ $C_{ovl} = 0$
Question 9	String	AN-Onset-Decoded vs Koickal-Decoded	PESQ = 2.1915 $C_{sig} = 0$ $C_{bak} = 0.5149$ $C_{ovl} = 0$
Question 10	Male Speech	Original vs AN-Onset-Decoded	PESQ = 1.9119 $C_{sig} = 1.478$ $C_{bak} = 1.8755$ $C_{ovl} = 1.6503$

TABLE 6.1: The Objective test scores on the sounds used in the Subjective test for question 1 to question 10. The better sound has been compared with the worse sound i.e. the better sound has been used as clean file and the worse one has been the enhanced file.

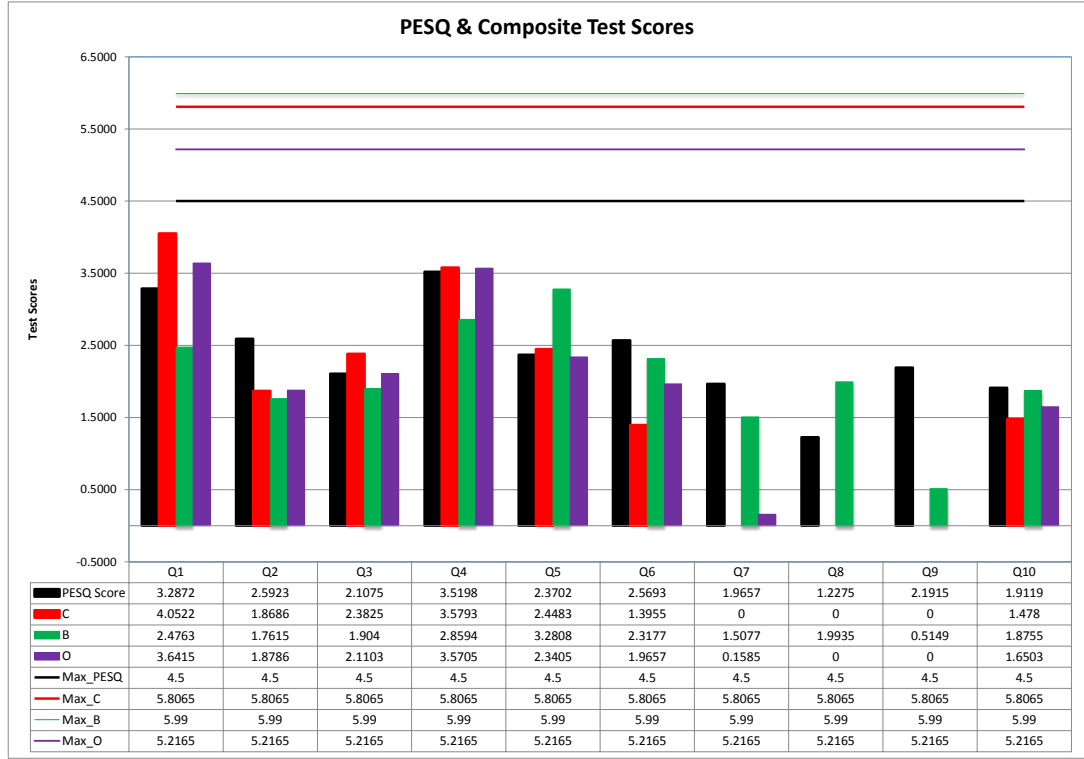


FIGURE 6.1: **The test scores of question 1 to 10:** The test scores have been plotted together and compared along with their maximum values. The C represents C_{sig} , B represents C_{bak} & O represents C_{ovl} ; as explained in chapter 2. The test scores 0 suggests that there is no similarity between the clean and enhanced sound at all. Originally MATLAB produces negative scores for some of them, but they have been normalized as 0 ([85]).

So in figure 6.1 and 6.2, we can find how much better the better sound is than the worse one (the sound which is ‘better’ than other has already been identified by the subjective test in table 5.41).

PESQ & Composite Test Scores			
Questions	Sound Type	Techniques Compared (Better vs Worse)	PESQ & Composite Test Score
Question 11	Percussion	AN-Decoded vs Koickal-Decoded	PESQ = 0.6075 $C_{sig} = 0$ $C_{bak} = 1.2157$ $C_{ovl} = 0$
Question 12	String	AN-Decoded vs AN-Onset-Decoded	PESQ = 3.4238 $C_{sig} = 2.2207$ $C_{bak} = 2.8433$ $C_{ovl} = 2.7628$
Question 13	Female Speech	Original vs AN-Onset-Decoded	PESQ = 1.8279 $C_{sig} = 0$ $C_{bak} = 2.0625$ $C_{ovl} = 0.4005$
Question 14	String	AN-Decoded vs AN-Onset-Decoded	PESQ = 3.1210 $C_{sig} = 4.285$ $C_{bak} = 3.7741$ $C_{ovl} = 3.6893$
Question 15	Female Speech	Original vs AN-Decoded	PESQ = 3.2844 $C_{sig} = 3.3764$ $C_{bak} = 2.8285$ $C_{ovl} = 3.3392$
Question 16	String	Original vs AN-Decoded	PESQ = 4.0440 $C_{sig} = 4.2697$ $C_{bak} = 2.9924$ $C_{ovl} = 4.0971$
Question 17	Male Speech	Original vs Koickal-Decoded	PESQ = 2.5246 $C_{sig} = 1.9257$ $C_{bak} = 2.5213$ $C_{ovl} = 2.2308$
Question 18	String	Original vs Koickal-Decoded	PESQ = 2.1212 $C_{sig} = 0$ $C_{bak} = 1.7872$ $C_{ovl} = 0.3814$
Question 19	Male Speech	AN-Decoded vs AN-Onset-Decoded	PESQ = 1.3569 $C_{sig} = 0$ $C_{bak} = 1.8293$ $C_{ovl} = 0.175$
Question 20	Female Speech	Original vs Koickal-Decoded	PESQ = 2.2416 $C_{sig} = 2.0587$ $C_{bak} = 2.346$ $C_{ovl} = 2.1434$

TABLE 6.2: The Objective test scores on the sounds used in the Subjective test for question 11 to question 20. The better sound has been compared with the worse sound i.e. the better sound has been used as clean file and the worse one has been the enhanced file.

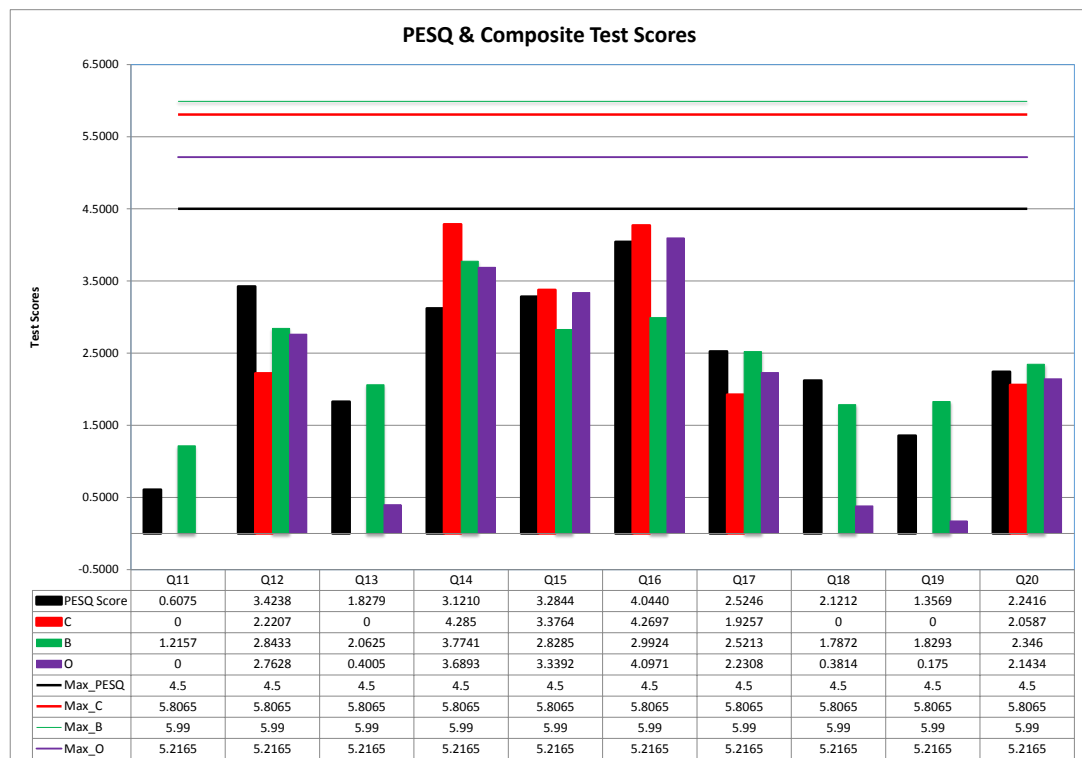


FIGURE 6.2: **The test scores of question 11 to 20:** The test scores have been plotted together and compared along with their maximum values. The test scores 0 suggests that there is no similarity between the clean and enhanced sound at all. Originally MATLAB produces negative scores for some of them, but they have been normalized as 0 ([85]).

6.4 Findings from PESQ on: the five sounds in subjective test

The composite PESQ test scores have been calculated on those five sounds: percussion, string 1 (celesta frequency 4), string 2 (celesta frequency 7), male voice and female voice regenerated by AN, AN-Onset and Koickal's spike codes to see which sound can be better coded by which spike coding techniques. Figure 6.3 explains that male voice has the highest PESQ score for AN-spike coding followed by lower frequency Celesta and the female speech. So, AN spike code is more successful for the male speech and lower frequency-musical sounds.

The AN-Onset spike code is good for male voice and the celesta frequency 4 musical note as well, and also for the female speech. Koickal's spike code is better for coding speech than coding other musical sort of sounds.

Overall AN coding is good for representing sounds of all sorts as the average score has been achieved as 3.13 where the highest score has been 3.51 (the maximum possible score is 4.5). Figure 6.4 also provides the evidence that the AN-spike coding is the best as it provides the best quality of decoded sound followed by the AN-Onset spike code and Koickal's spike code. In the following figures, 'PESQ-Thomas', 'C-Thomas', 'B-Thomas' and 'O-Thomas' represents the scores from Koickal's spike coding.

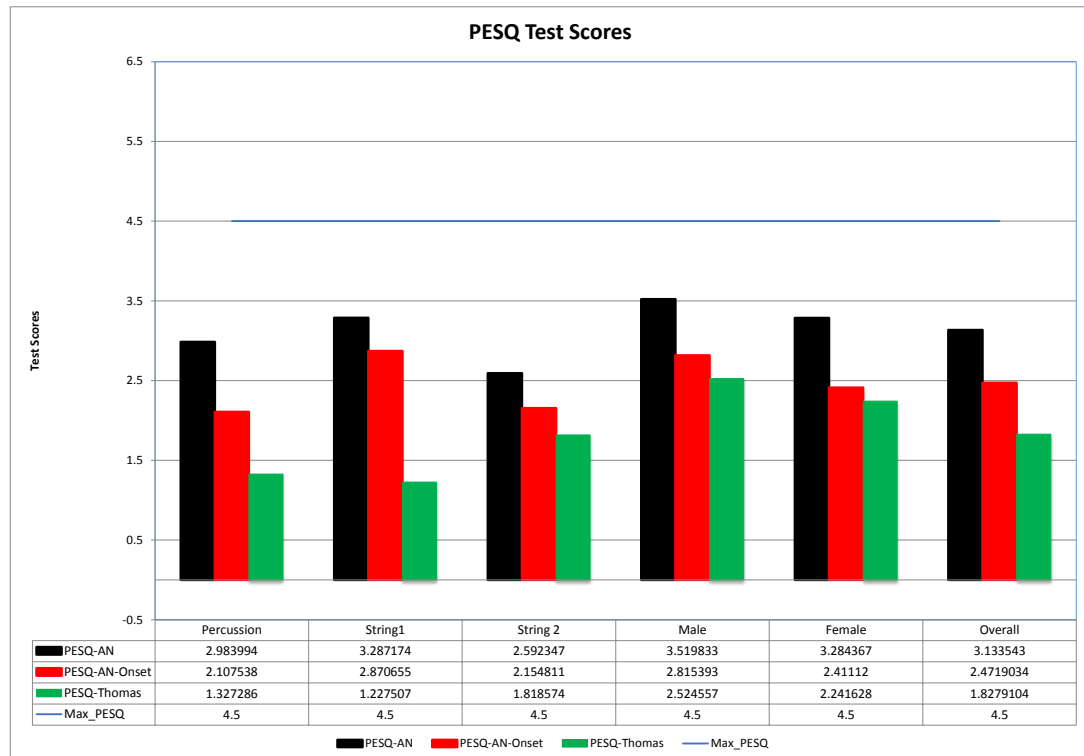


FIGURE 6.3: PESQ test scores for **different types of sounds**: Here, the PESQ scores have been compared for the five different sounds and they are shown in this figure according to those five different sound types. ‘PESQ-Thomas’ represents the PESQ scores from Koickal’s code. AN & AN-Onset spike codes produce the highest PESQ test scores for Male Voice and the Celesta Frequency 4 (String1). So, the Male Voice and the Celesta Frequency 4 (String1) has been decoded as the best among others for AN and AN-Onset spike coding.

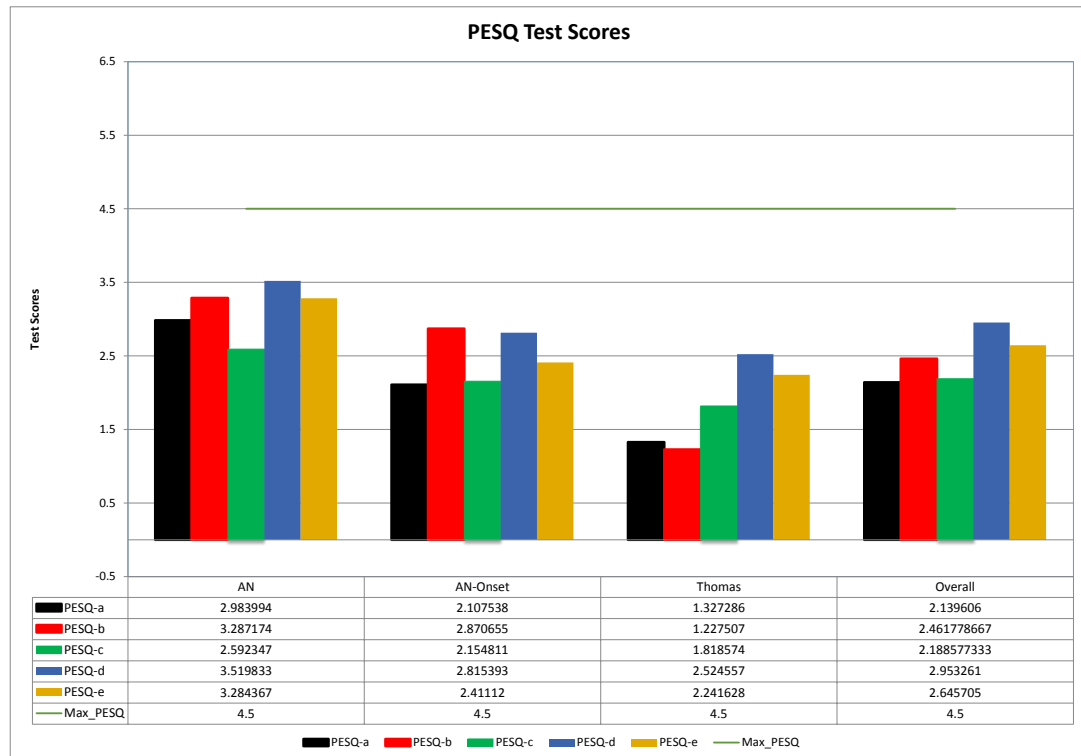


FIGURE 6.4: PESQ Test Scores for **different types of codes**: Here, the PESQ scores have been compared for the five different sounds and they are shown in this figure according to three different spike coding techniques. Here, ‘Thomas’ represents Koickal’s code. PESQ-a represents ‘Percussion’ sound, ‘b’ represents ‘String 1’, ‘c’ represents ‘String 2’, ‘d’ represents ‘Male Speech’ & ‘e’ represents ‘Female Speech’. The PESQ scores suggest that AN spike code is the best, AN_Onset spike code is better and Koickal’s spike code is the worse for all of these five different types of sounds.

6.5 Findings from PESQ on: Number of Channels and Sensitivity levels

As mentioned at section 3.5 in chapter 3, the number of channels and sensitivity levels are an important factor in spike coding and decoding of a sound. Initially, the number of channels have been set as: 50 and sensitivity levels as: 16. Lowering these numbers makes the spike coding more lossy which leads to very low quality of regenerated sounds. But how low the quality of those reconstructed sounds is compared to the original? This section provides some evidence of choosing the right number of channels and sensitivity levels based on the high composite PESQ test scores and the amount of bytes required to store the spikes.

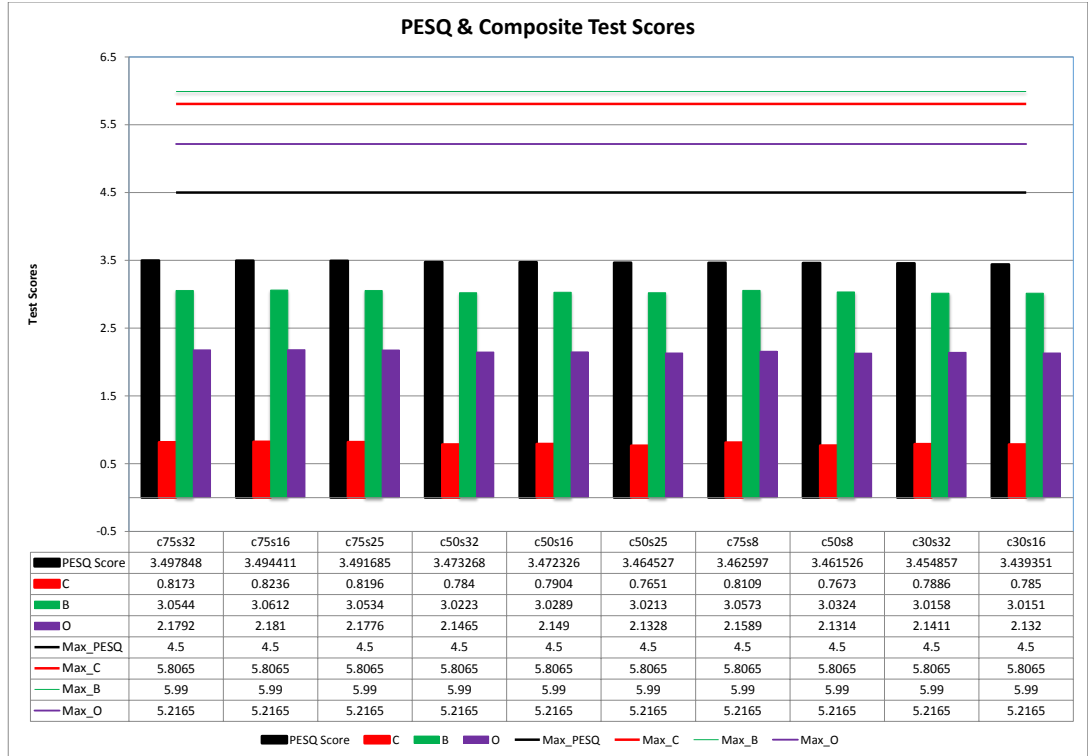


FIGURE 6.5: The PESQ test scores of different numbers of channels and sensitivity levels ($c_{N s_M}$ means total N channels and M sensitivity levels are used) for BEST PESQ: The composite PESQ test scores have been plotted together and compared along with their maximum values. This figure has been sorted according to the highest PESQ score. The test scores 0 suggests that there is no similarity between the clean and enhanced sound at all ([85]).

Figure 6.5 shows that 75 channels and 32 sensitivity levels have the highest PESQ scores which decreases along with the number of sensitivity levels. However 8 sensitivity levels

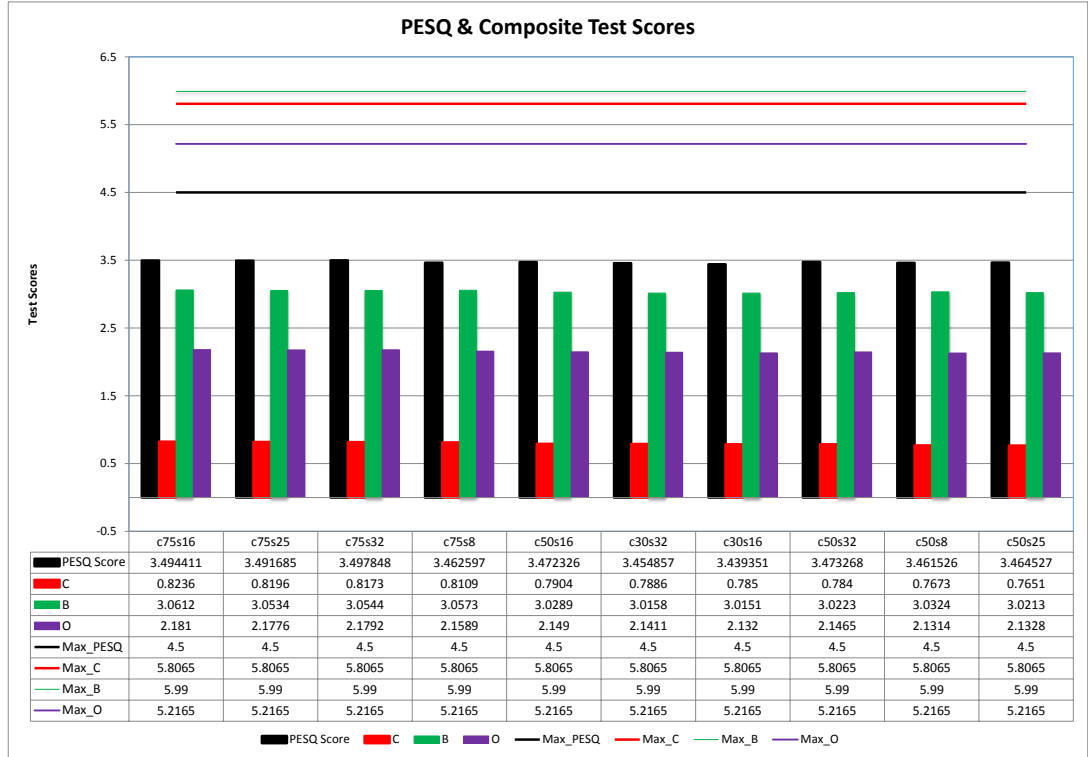


FIGURE 6.6: The PESQ test scores of different channels and sensitivity levels (c_{NSM} means total N channels and M sensitivity levels are used) for BEST Signal Distortion (C_{sig}): The composite PESQ test scores have been plotted together and compared along with their maximum values. This figure has been sorted according to the highest C_{sig} score. The test scores 0 suggests that there is no similarity between the clean and enhanced sound at all ([85]).

do not produce a good quality of regenerated sound so they lie towards the end of figure 6.5. So, it is evident from this figure 6.5, that lowering the number of channels and sensitivity levels decreases the quality of speech.

Figure 6.6 shows that 75 channels and 16 (not 32) sensitivity levels, has the highest C_{sig} scores which decreases with the number of channels. From that figure we can say that the number of channels is a major factor, but differences are rather small, in the measure of signal distortion in a speech. More channels means less signal distortion.

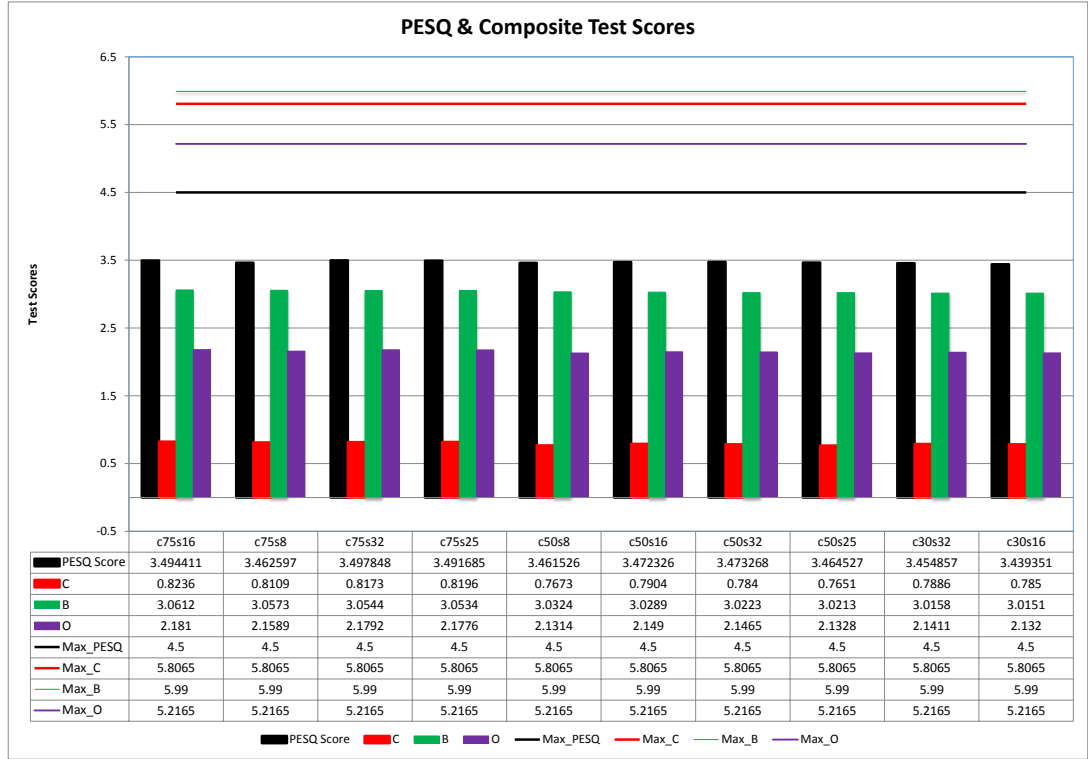


FIGURE 6.7: The PESQ test scores of different channels and sensitivity levels (c_{NSM} means total N channels and M sensitivity levels are used) for BEST Noise Distortion (C_{bak}): The composite PESQ test scores have been plotted together and compared along with their maximum values. This figure has been sorted according to the highest C_{bak} score. The test scores 0 suggests that there is no similarity between the clean and enhanced sound at all ([85]).

Figure 6.7 shows that 75 channels and 16 (not 32) sensitivity levels, has the highest C_{bak} scores which decreases with the number of channels. From this figure we can say that the number of channels is the major factor, again with small differences, in the measure of noise distortion in a speech. More channels means less noise distortion.

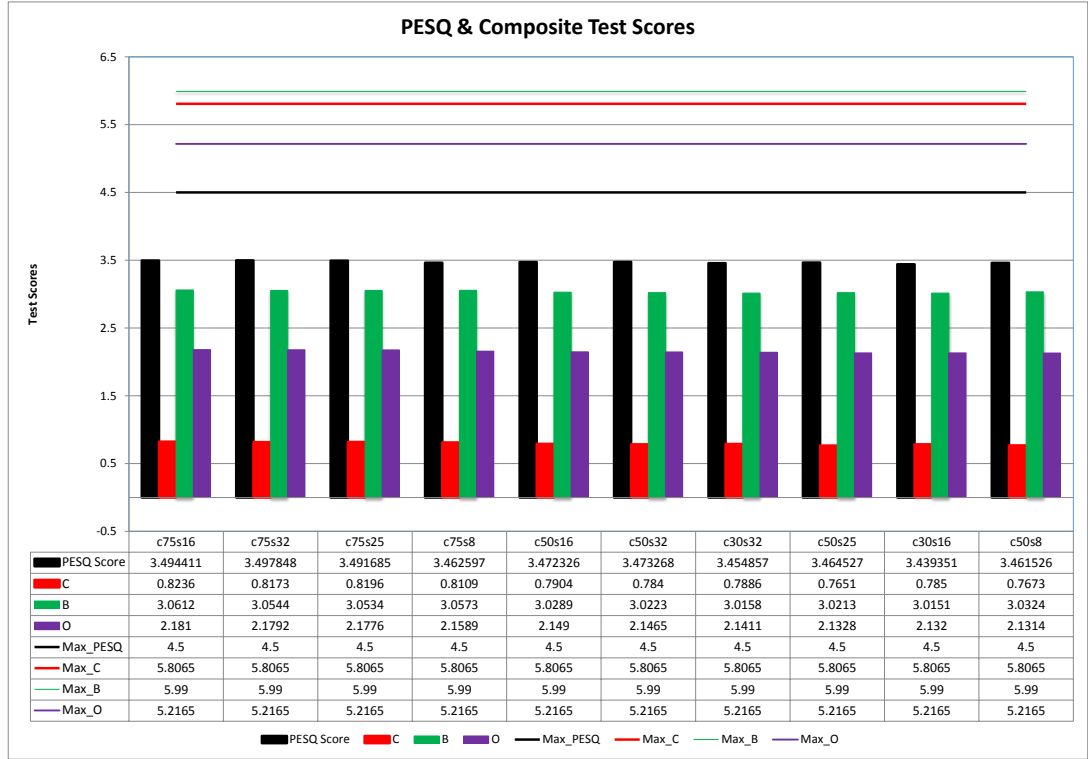


FIGURE 6.8: The PESQ test scores of different channels and sensitivity levels (c_{NSM} means total N channels and M sensitivity levels are used) for BEST C_{ovl} : The composite PESQ test scores have been plotted together and compared along with their maximum values. This figure has been sorted according to the highest C_{ovl} score and the differences are really quite small. The test scores 0 suggests that there is no similarity between the clean and enhanced sound at all ([85]).

Figure 6.8 shows that 75 channels and 16 (not 32) sensitivity levels, has the highest C_{ovl} scores which decreases almost with the number of channels. Towards the end, channel 30 with sensitivity level 16 has better similarity overall with the original sounds than channel 50 and sensitivity level 8. From this figure and previous two, we can say that the number of channels is the most important factor in the measure of overall quality in a speech. More channels with a sufficient number of sensitivity level means better quality in regenerated speech.

We note that the PESQ test scores do not differ too much from each other. This indicates that regardless of the number of channels and sensitivity levels, a good quality of sound can be reconstructed. However, the decisions have been taken by comparing the slightest differences in those PESQ test scores.

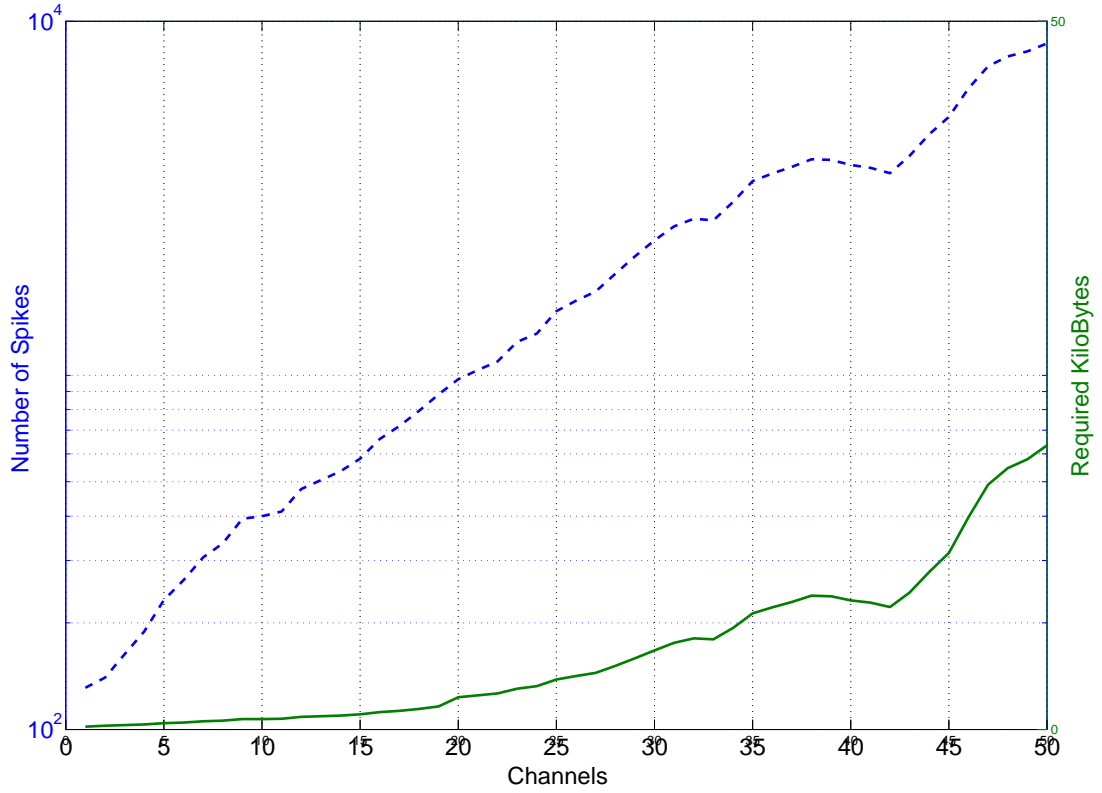


FIGURE 6.9: **The bytes required to store AN spikes for 16 sensitivity levels & 50 channels:** This figure plots the required kilobytes along with the number of spikes for each channel. On the left side, the number of spikes are shown which ranges from 10^2 to 10^4 and on the right side, the required kilobytes are shown which ranges from 0 to 50 kilobytes. For a 2.8706 second sound file, 274.7018 kilobytes of AN spikes are required at 16 sensitivity levels and 50 channels.

This figure 6.9 explains how many spikes have been generated and how much space does it require to be stored in database. The number of bytes required has been calculated based on the number of spikes in a channel (I_{ch}) and the channel frequency (f_{ch}). The level of accuracy to code a spike at channel ch has been Λ decimal places, where $\Lambda = \lceil n \rceil$, where $n = \log_{10}(10 \times f_{ch})$.

The required bits to code all the spikes in that channel ch is $I_{ch} \times P$, where $P = \lceil p \rceil$ and $p = \log_2(T \times 10^\Lambda)$. (Here, T is the time length of the sound in seconds.)

The Total kilobytes to code all the spikes for the sound is : $\frac{\sum_{ch=1}^N (I_{ch} \times P)}{8 \times 1024}$, where 1 Byte = 8 bits & 1 kilobyte = 1024 Bytes.

Let's discuss it by an example. Say, for a channel frequency 123.8965, the number of spikes are 140. So, I_{ch} is 140 and f_{ch} is 123.8965. n will be $\log_{10}(10 \times 123.8965) = 3.0931$ and Λ will be 4, i.e. the level of accuracy will be 1 in 10000.

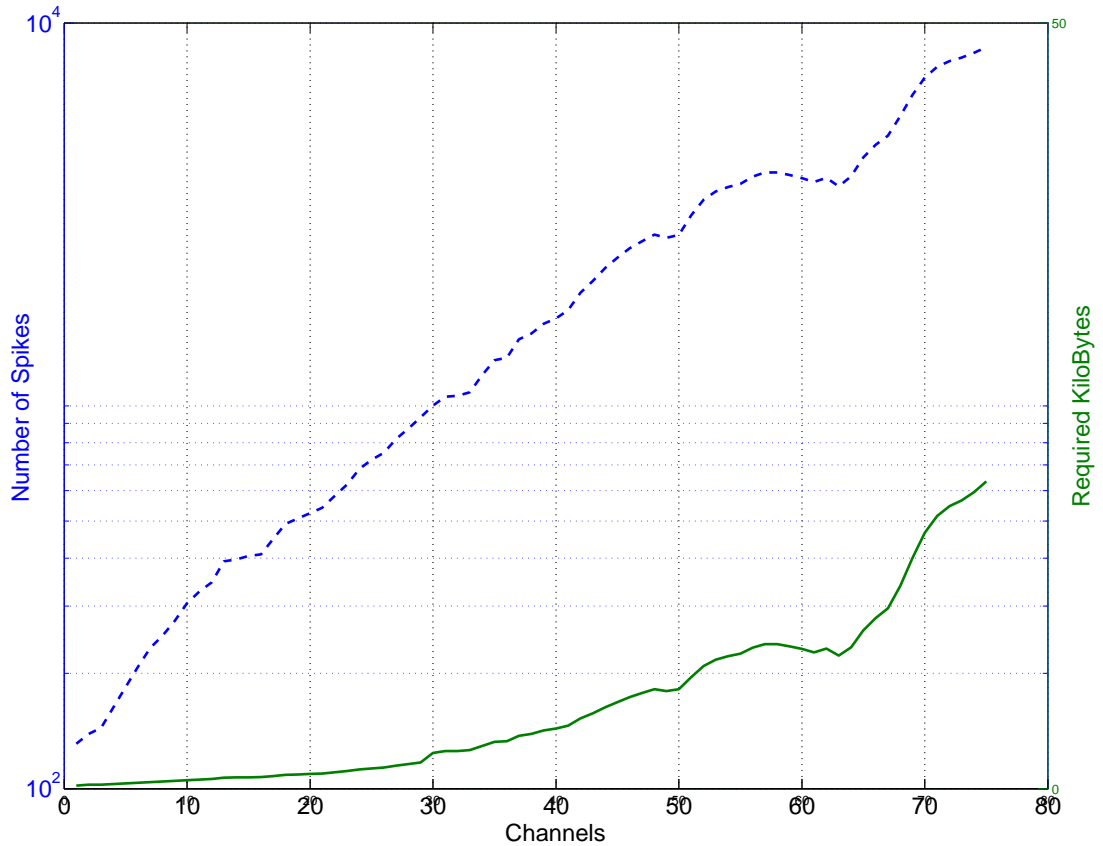


FIGURE 6.10: **The bytes required to store AN spikes for 32 sensitivity levels & 75 channels:** This figure plots the required kilobytes along with the number of spikes for each channel. Bytes required for AN spikes of a 2.8706 sec sound file is 409.2827 kilobytes at 32 sensitivity levels and 75 channels.

So, p will be $\log_2(2.8706 \times 10000) = 14.8091$, as the sound file is 2.8706 sec long. P will be 15. So, the number of bits required to code all the spikes in that channel will be $140 \times 15 = 2100$. Considering all the spikes in all the channels the total kilobytes required is 274.7018 KB.

This figure 6.9 and 6.10 explains how many spikes have been generated and how much space is required for storing and we can definitely say that more channels and sensitivity levels produces more spikes which makes the spike coding less lossy. However the number of them and the required space has to be considered. For 50 channels and 16 sensitivity levels, only 275 kilobytes are required but for 75 channels and 32 sensitivity levels, 409 kilobytes are necessary. However both produces reasonably well regenerated sound and for that reason, channel number 50 and sensitivity level 16 has been chosen as the most effective number of channels and sensitivity levels.

6.6 Findings from PESQ on Onset Parameters

As explained at equation 2.5, equation 2.6 and equation 2.7 in chapter 2; the values of the parameters are important to generate onset spikes for each significant increase in the energy of the sounds by leaky integrate-and-fire neurons. Initially, the parameter values have been set as $\alpha = 100$, $\beta = 9$ and $g = 1100$, where these parameters set the characteristics of the depressing synapse. And the other ‘rp_wide’ = 0.0015, ‘onset_cell_weight (weight of each cell)’ = 1000 and ‘spread_wide (number of AN fibers on each side of the cell)’ = 1. These values have been changed to see which combination of parameters provide the best Onset coding technique which will be able to decode the best quality of sound. The combinations are as follows

Combination 1:

$\alpha = 500$, $\beta = 25$, $g = 1100$ (for depressing synapse); ‘rp_wide’ = 0.0015, ‘onset_cell_weight’ = 1000 and ‘spread_wide’ = 6

Combination 2:

$\alpha = 500$, $\beta = 25$, $g = 1100$ (for depressing synapse); ‘rp_wide’ = 0.0015, ‘onset_cell_weight’ = 1000 and ‘spread_wide’ = 1

Combination 3:

$\alpha = 100$, $\beta = 9$, $g = 1100$ (for depressing synapse); ‘rp_wide’ = 0.0015, ‘onset_cell_weight’ = 500 and ‘spread_wide’ = 4

Combination 4:

$\alpha = 100$, $\beta = 9$, $g = 1100$ (for depressing synapse); ‘rp_wide’ = 0.003, ‘onset_cell_weight’ = 1000 and ‘spread_wide’ = 6

Combination 5:

$\alpha = 100$, $\beta = 9$, $g = 1100$ (for depressing synapse); ‘rp_wide’ = 0.003, ‘onset_cell_weight’ = 500 and ‘spread_wide’ = 4

So, a speech has been coded by Onset spike coding technique with those different combination of parameters and the PESQ and composite test scores have been discussed in figure 6.11.

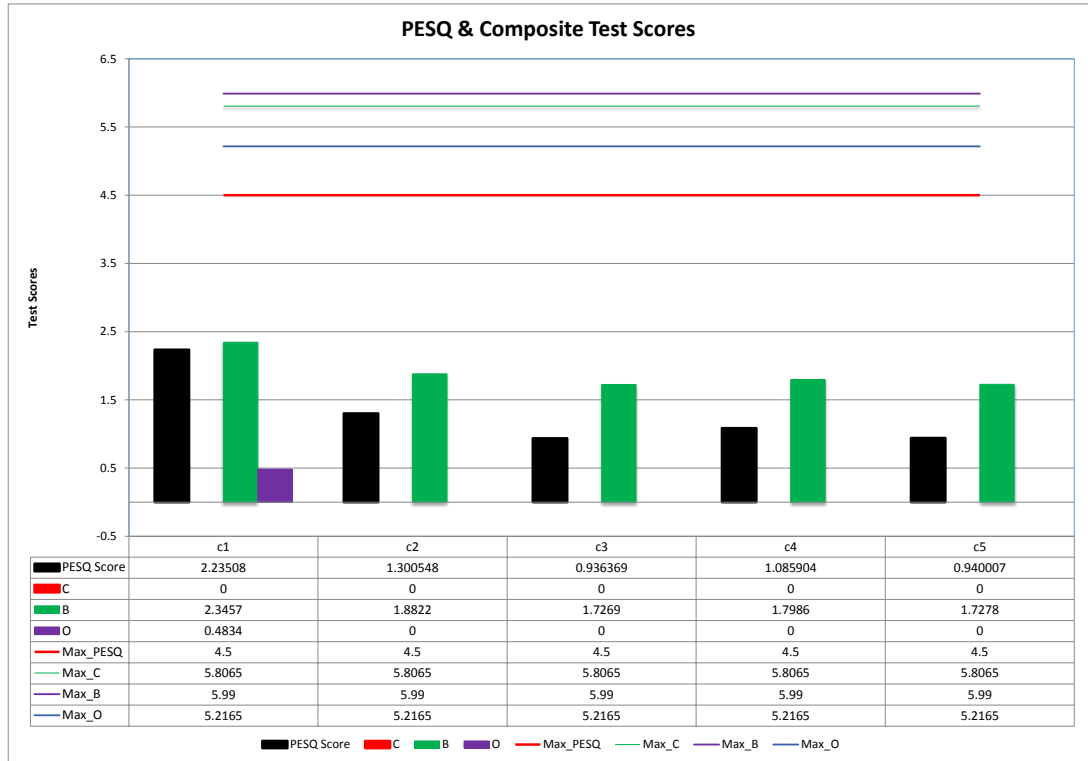


FIGURE 6.11: PESQ test scores for **different onset parameters**: Here the PESQ and composite scores for 5 different onset-parameter combinations (c1 to c5) are used. The test scores show that combination 1 provides the highest PESQ scores indicating that using that combination of parameter values in coding Onset spikes will provide the best quality of decoded sound.

According to the figure 6.11 where the decoded speech has been tested with the original with those 5 different combinations of parameter values, we can see that first combination provides the highest PESQ test score. The signal distortion for all of them has been negative, so zero has substituted those negative values ([64]). This indicates that the signal has been distorted from the original sound very much for all of those parameter values. So, it can be concluded that Onset spike coding cannot provide a decoded signal without distorting the original sound. Overall the composite score indicates that the first combination [$\alpha = 500$, $\beta = 25$, $g = 1100$ (for depressing synapse); ‘rp_wide’ = 0.0015, ‘onset_cell_weight’ = 1000 and ‘spread_wide’ = 6] provides the best decoded sound for Onset spike coding technique. The corresponding PESQ score for the combination 1 has been 2.34, whereas the other scores are well below than 2.34. So, we can conclude that using combination 1 in onset spike coding technique will provide the best quality of decoded sound.

6.7 Findings from PESQ on: Resolution Bits in Koickal's Spike Coding

In Koickal's [5] spike code, the resolution bits are important parameter to code the sound efficiently. As mentioned at 2.3.3 in 2, the Resolution parameter (δ) has been set to '7'. Here we explain why this has been set to that particular value.

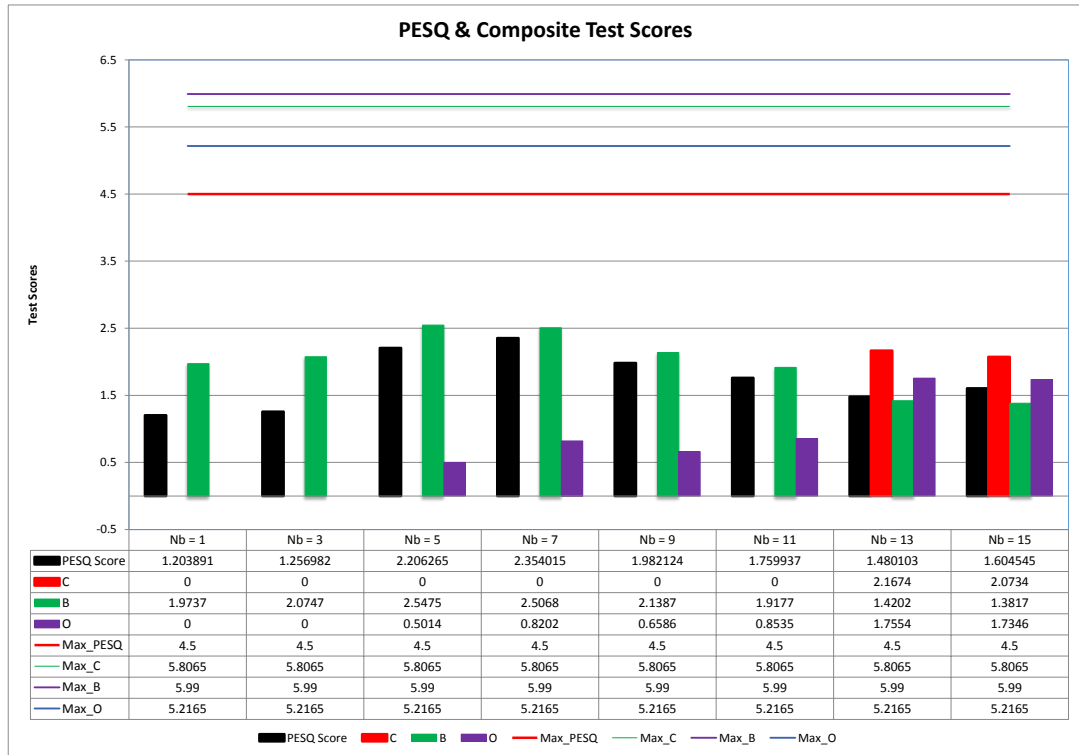


FIGURE 6.12: **PESQ test scores for different parameters in Koickal's sound reconstruction:** This figure shows that for the resolution parameter value '7', the PESQ score is the highest. And also, the other composite scores are high for resolution value '7' as well. So, the Resolution parameter value has been set to '7'.

Figure 6.12 explains why the resolution parameter can be set to '7'. For the value '7', the PESQ score is 2.35 which is the highest among all the other values. The Resolution parameter '5', the PESQ score is 2.21 which is quite close to 2.35. So, a good quality of sound can still be produced with the value '5'. However the number of spikes generated for those two cases are quite different from each other. For 5 bits, there are 113281 spikes and for 7 bits, there are 129933 spikes. So, by increasing the resolution bits from 5 to 7, number of spikes increases by 14.17% & the PESQ score increases by 6.33%.

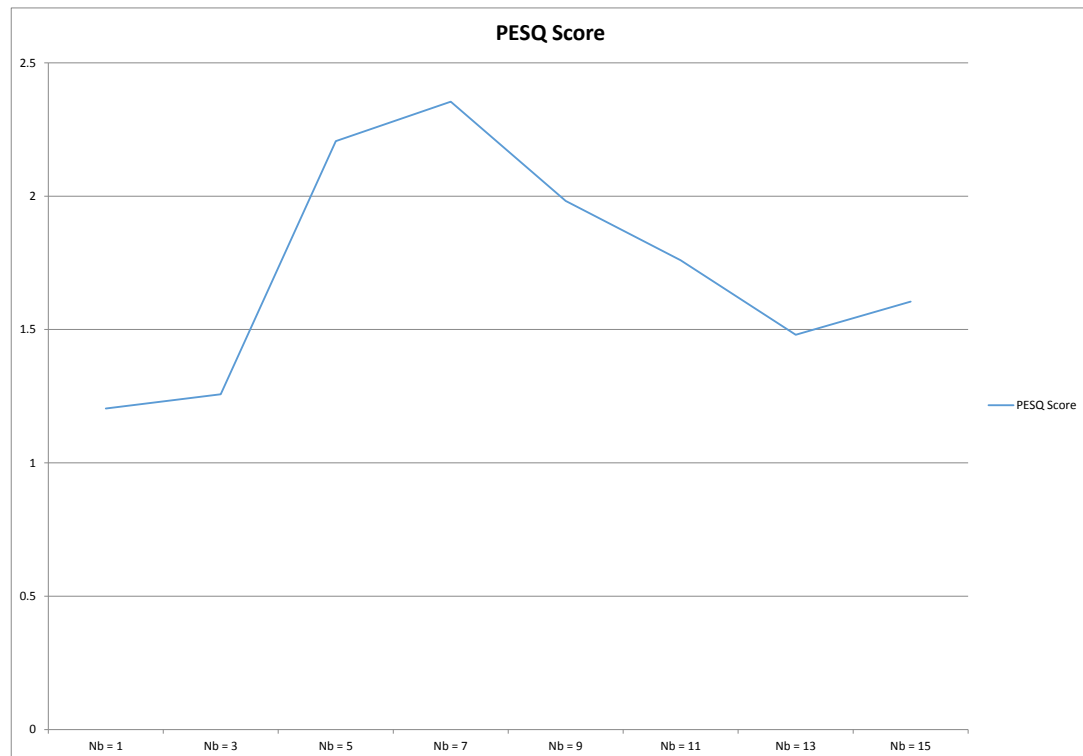


FIGURE 6.13: **PESQ test scores for different parameters in Koickal’s sound reconstruction:** This figure shows that for the resolution parameter value -‘7’, the PESQ score is the highest. So, the resolution parameter value has been set to ‘7’.

So, for the resolution bits 5 and 7, Koickal’s spikes provide the best quality of regenerated sound than any other resolution bits. The figure 6.13 shows the graph of the PESQ scores only for all the resolution bits used in this test.

6.8 Findings from PESQ on: Musical Notes with Different Frequencies

Although PESQ is primarily intended for the speech type of signal, the musical type of sounds has been compared by PESQ as well. Five musical instruments were played at different tone and frequency and they have been coded and decoded by the AN, AN-Onset and Koickal’s spike coding technique. Now, the decoded sounds have been compared with the original sound and the corresponding PESQ scores have been obtained. They have been discussed in detail below.

6.8.1 Celesta

The Celesta has been played at A4, A5, A6 & A7 tones. A4 is for the lower frequency and A7 is for the higher frequency. Figure 6.14 shows that there is a rapid decline among the PESQ scores based on the increase of the frequency for the AN type of spike coding. However this is not true for the AN-Onset type of coding and Koickal's spike coding. Their values do not follow any particular pattern. The standard deviations for

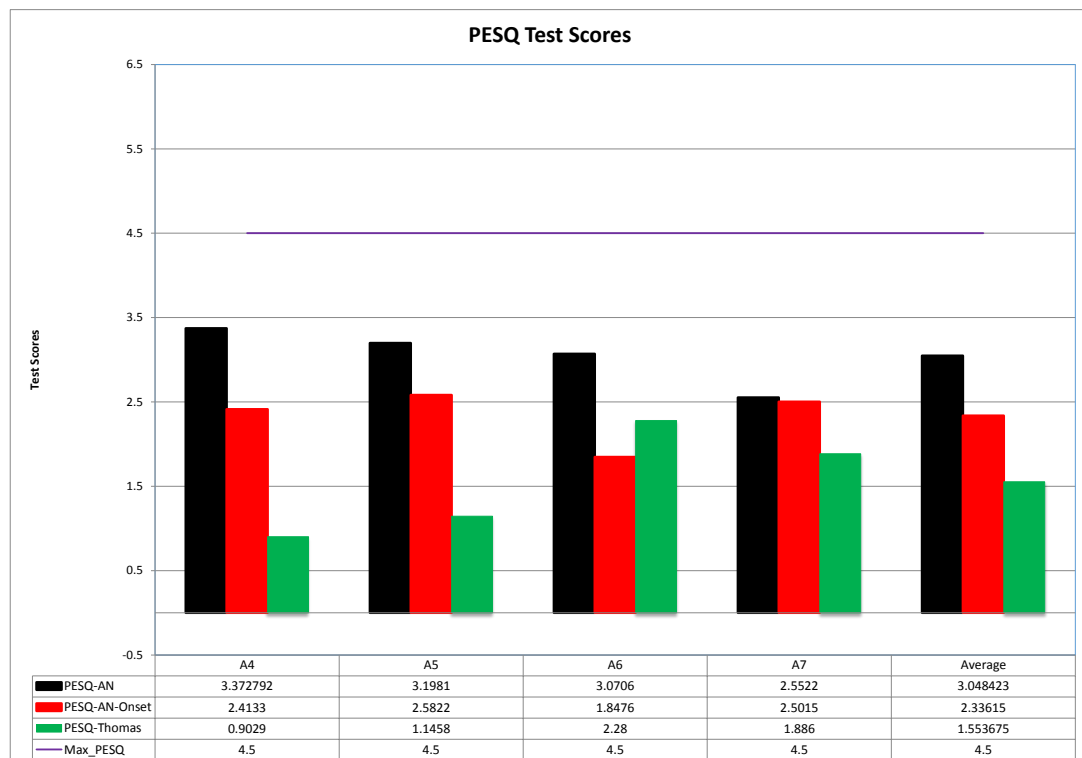


FIGURE 6.14: **PESQ test scores for Celesta notes:** This figure shows the PESQ scores for Celesta notes A4 to A7. ‘PESQ-Thomas’ represents the PESQ scores from Koickal’s code.

those PESQ scores are: 0.306 for AN coding, 0.288 for AN-Onset coding and 0.554 for Koickal’s spike coding. So, Koickal’s coding has the maximum standard deviation for Celesta musical notes. The AN coding works the best for lower frequency of Celesta musical notes.

6.8.2 Harp

Next, the Harp has been played for the Musical notes A3 to A5. Figure 6.15 discusses the PESQ scores for those musical notes. Interestingly for the Harp type of sound, we can find the increase of PESQ scores with the increase of note frequency for AN spike coding; which is completely different from the Celesta and Gemshorn. The AN-Onset coding also has a sharp steady decline with the increase of frequency notes which is the complete opposite to the AN spike coding. The standard deviations are: 0.27 for AN, 0.22 for AN-Onset & 0.16 for Koickal's spike code. So, the PESQ scores didn't change much throughout the frequency notes.

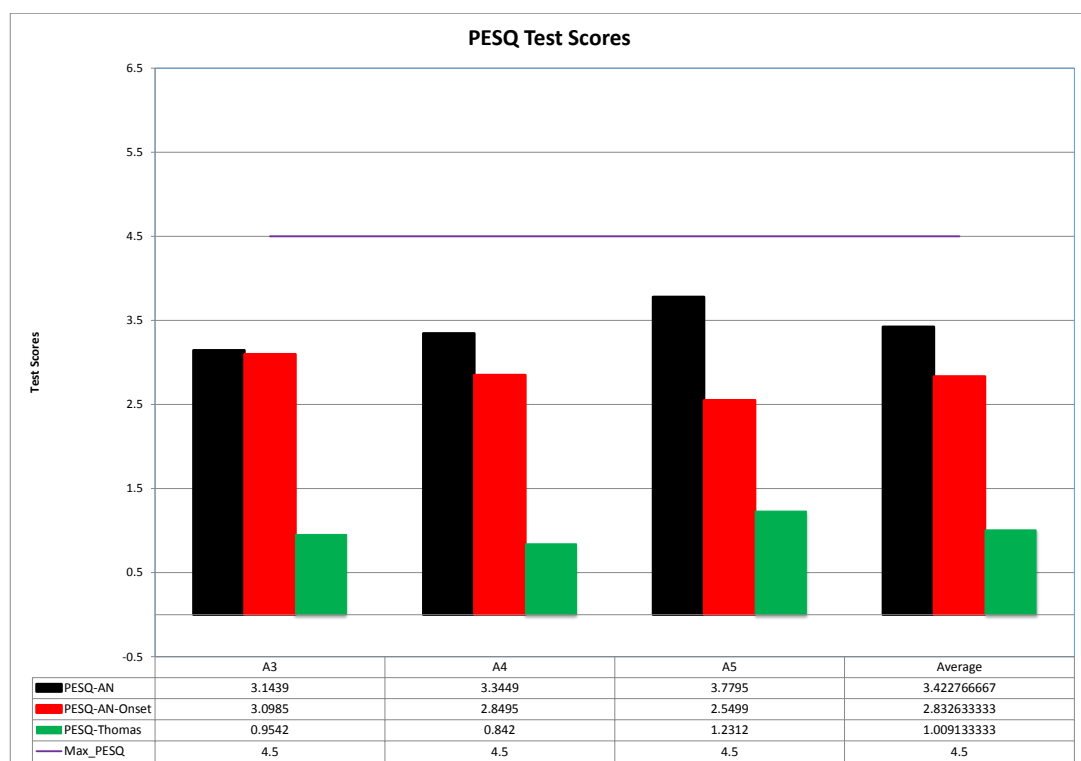


FIGURE 6.15: **PESQ test scores for Harp notes:** This figure shows the PESQ scores for Harp notes A3 to A5. 'PESQ-Thomas' represents the PESQ scores from Koickal's code.

6.8.3 Guitar

Guitar has been played similarly like before and has been shown in the figure 6.16. For Guitar, we can find similarity with Gemshorn. However the standard deviation has been really low for AN coding: 0.138 & for Koickal's coding, it has been as high as 0.5.

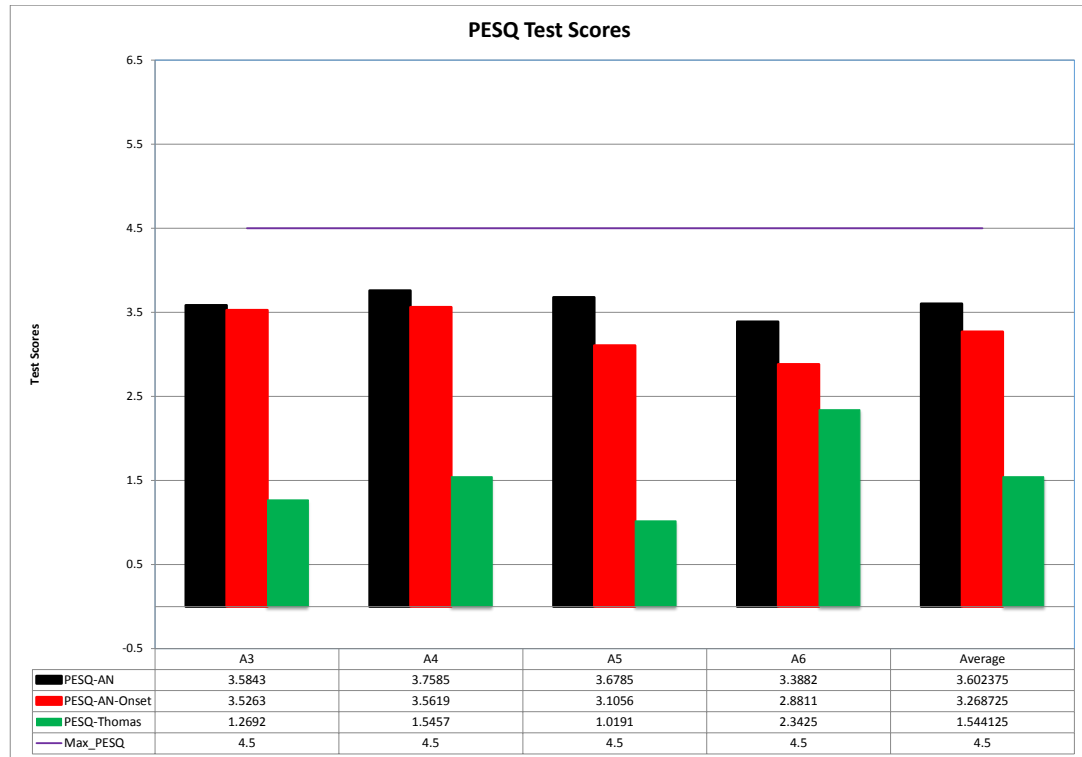


FIGURE 6.16: **PESQ test scores for Guitar notes:** This figure shows the PESQ scores for Guitar notes A3 to A6. ‘PESQ-Thomas’ represents the PESQ scores from Koickal’s code.

6.8.4 Pianos

There are several notes from Piano have been played and then the regenerated signals have been put through the PESQ test scores. The scores has been mentioned in the figure 6.17. AN spike coding has produced good quality of regenerated notes for most of them except B4. For that note B4, the AN-Onset code has generated a higher score than AN code. However, it is difficult to find out a pattern among the scores as most of them are pretty well. Koickal's spike code hasn't been good for very low frequency note: B0 & B1. The standard deviation of Koickal's spike coding has been really high: 0.71; whereas for AN spike coding, it is 0.21 & AN-Onset is 0.39.

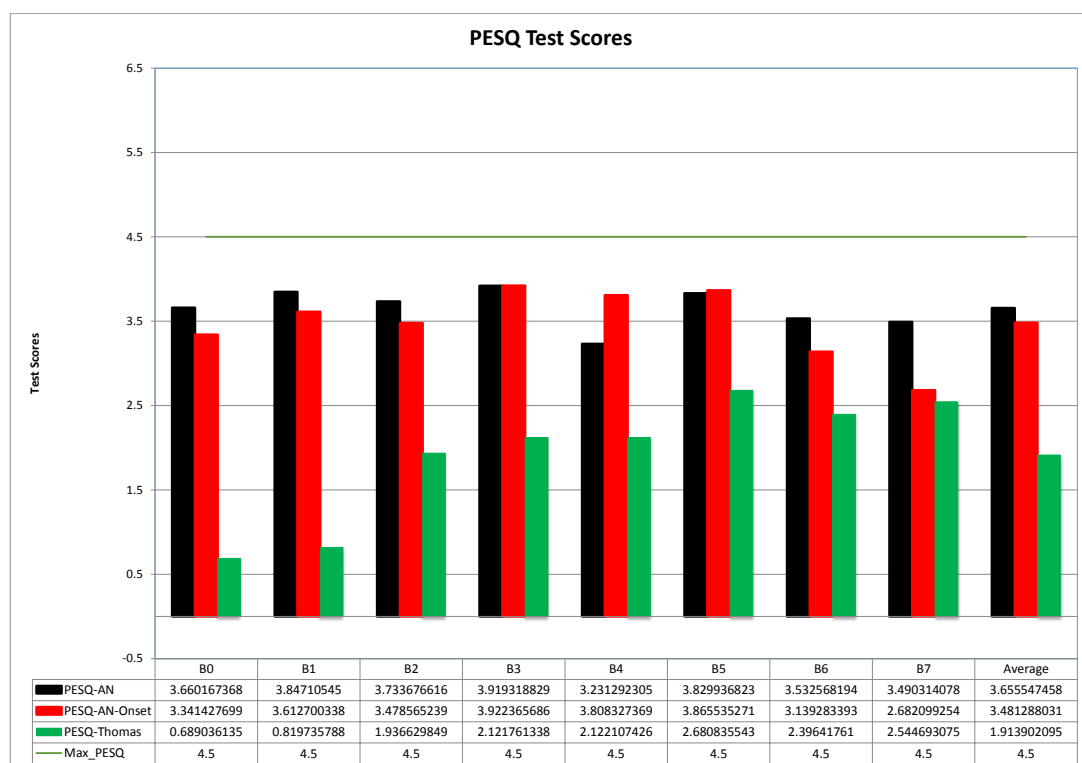


FIGURE 6.17: **PESQ test scores for Piano loud notes:** This figure shows the PESQ scores for piano loud notes from B0 to B7. ‘PESQ-Thomas’ represents the PESQ scores from Koickal’s code.

6.8.5 Woodwinds

Few frequency notes of Woodwinds have been tested by PESQ and the scores have been mentioned in figure 6.18. The standard deviations have been really low such as: 0.08 for AN, 0.04 for AN-Onset & 0.12 for Koickal's code.

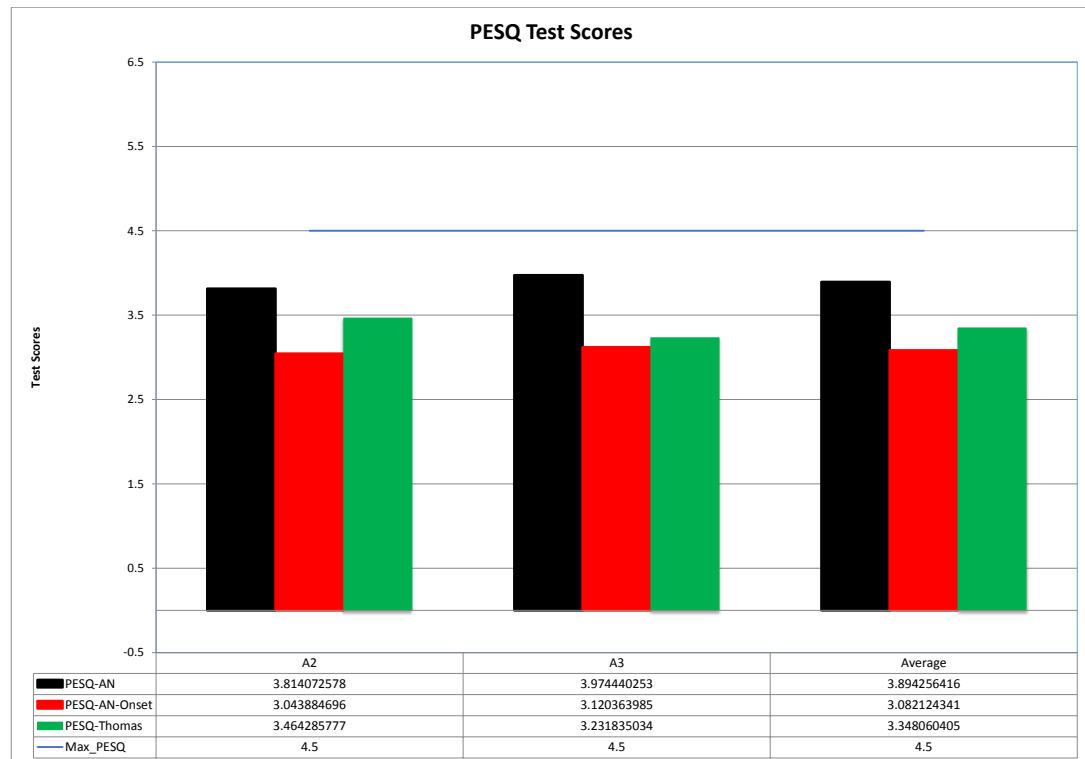


FIGURE 6.18: **PESQ test scores for Woodwinds notes:** This figure shows the PESQ scores for woodwinds notes for A2 & A3. 'PESQ-Thomas' represents the PESQ scores from Koickal's code.

6.8.6 Gemshorn

Gemshorn has been played with different frequency notes and the PESQ scores have been displayed in figure 6.19. Here, AN coding worked the best for the A4 note and then gradually decreases with the change of frequency. There is a straight forward pattern in the PESQ scores for AN-Onset coding. It also worked the best for A4 note and then decreased with the change of frequency. However Koickal's code worked best for the A3 note and then sharply decreased with the increase of note frequency. The standard deviation for them stands as: 0.398 for AN, 0.745 for AN-Onset & 0.326 for Koickal's Code. So, we can find rapid changes in the AN-Onset spike coding quality for various musical notes frequency.

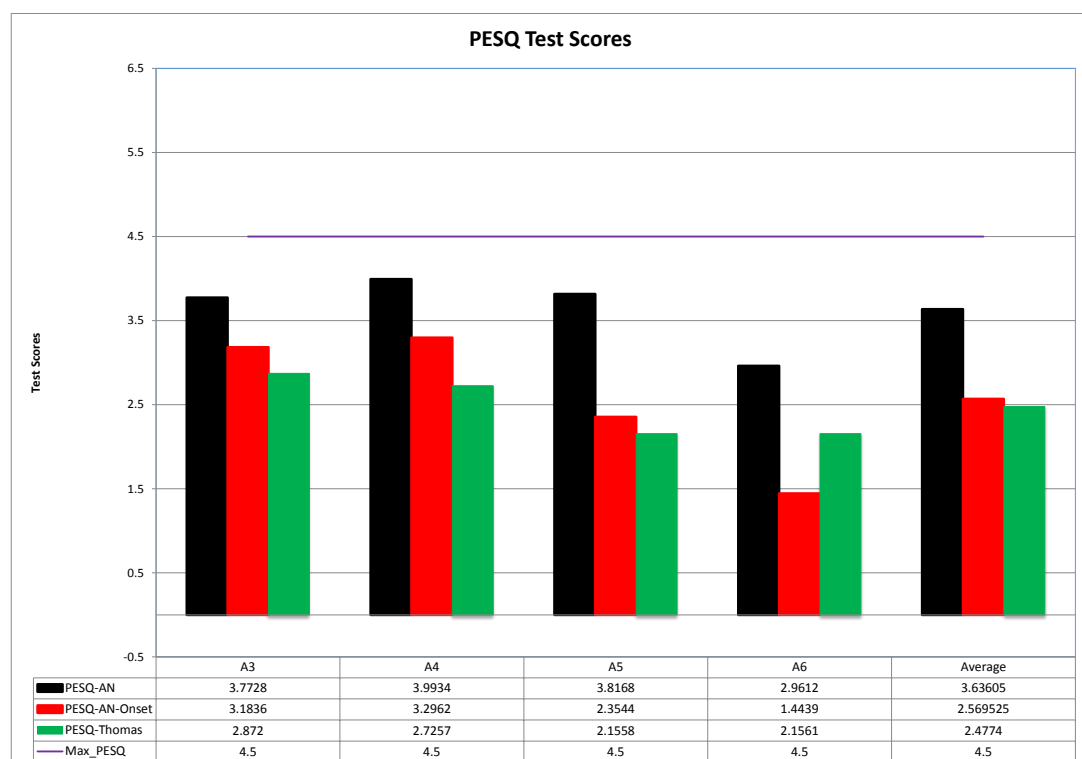


FIGURE 6.19: **PESQ test scores for Gemshorn notes:** This figure shows the PESQ scores for Gemshorn notes A3 to A6. 'PESQ-Thomas' represents the PESQ scores from Koickal's code.

The number of spikes has also been investigated to see the effectiveness of each spike coding technique.

So, according to table 6.3, we can say that the results mentioned in section 5.9 at table 5.42 are true for other sorts of musical notes as well. AN spikes are always highest

Number of Spikes for Gemshorn Sounds				
Sound	Sound Length	AN	AN_Onset	Koickal
Gemshorn at frequency 3	7.138 sec	765392	111884	407542
Gemshorn at frequency 6	7.4012 sec	862943	115950	482213

TABLE 6.3: **Number of spikes** : Here the number of spikes for two different frequencies of Gemshorn tunes have been mentioned along with the corresponding length in seconds. The numbers for each coding type are the total number of spikes used to decode the reconstructed sound in that coding technique. So, to decode the sound from AN_Onset spikes codes, 111884 spikes have been used for frequency 3 Gemshorn sound.

in numbers and AN_Onset spikes are lowest in numbers. So, the conclusion drawn in section 5.9 is valid for all sorts of musical sounds.

PESQ Test Scores	AN	AN_Onset	Koickal
Average	3.543	2.928	1.974
Standard Deviation	0.26	0.395	0.757

TABLE 6.4: **PESQ test scores for all musical notes:** - These are the average PESQ test scores for all the Musical Notes along with their standard deviations.

So, in conclusion we can say that overall AN coding has been better to represent the musical notes as the average PESQ scores for all the musical notes has been 3.543 according to table 6.4.

6.9 Findings from PESQ on: a Choir

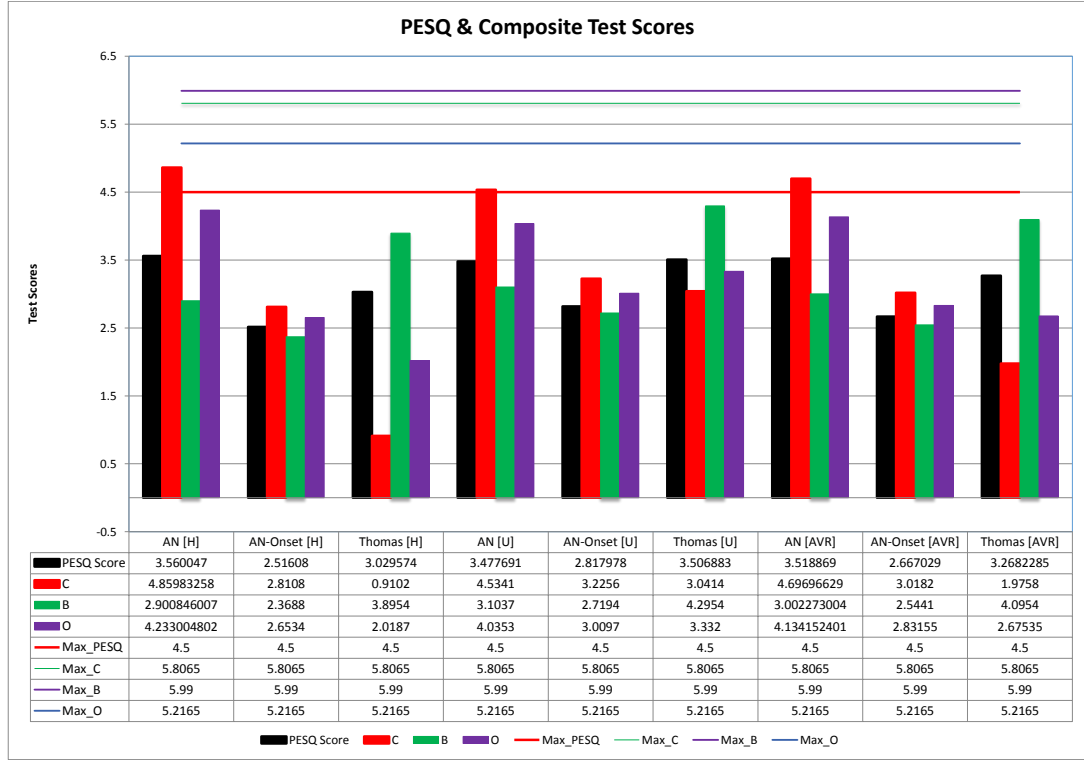


FIGURE 6.20: PESQ test scores for **Choir** type of sound: AN[H] represents the ‘hallelujah’ and AN[U] represents the choir song. AN[AVR] represents the average of those two. ‘Thomas[H]’ represents the PESQ scores from Koickal’s code for ‘hallelujah’ etc.

We have considered two choir sounds - one is ‘hallelujah’ in a Church and another one is small part of a song sung as a choir by many volunteers. Now from the figure 6.20, we can say that the PESQ scores are 3.56 & 3.48, not too far from each other for AN spike coding for both of them. Similarly, AN-Onset coding produces the decoded signals whose quality is not much different from the original sound either. However, for Koickal’s spike coding technique produces the decoded signals which gets the PESQ score value 3.03 & 3.51; which are a bit far away from each other. However it is not much significant as the average of them has been 3.27. The similar trend can be found among the composite scores as well not falling far away from each other. So, we can conclude that the spike coding technique is quite good for the choir type of sounds and they can work well for all sorts of choir sounds.

Also, as usual the AN spike code produces quite good quality of decoded sound for choir songs. However the AN-Onset spike code has fallen behind the Koickal’s spike code in

this case as the average PESQ score has been 2.66 for AN-Onset spike code, but it has risen to 3.26 for Koickal's spike code. So, this is a fact that Koickal's spike code is better than the AN-Onset spike code for choir type of sounds.

Number of Spikes for Choir Sounds				
Sound	Sound Length	AN	AN_Onset	Koickal
'hallelujah' in a Church	10.3366 sec	1138192	160274	575659
Choir Song	8.5265 sec	1084753	134282	472257

TABLE 6.5: **Number of spikes** : Here the number of spikes for two different Choir sounds have been mentioned along with the corresponding length in seconds. The numbers for each coding type are the total number of spikes used to decode the reconstructed sound in that coding technique. So, to decode the sound from AN_Onset spikes codes, 160274 spikes have been used.

Also, from this table 6.5, we can say that AN spikes are less lossy so they are many in numbers. Koickal's spikes are also much higher than AN_Onset spikes. However, from figure 6.20, we can see that Koicka's decoded sound is rather better than AN_Onset spikes. So, AN_Onset spike code cannot outperform Koickal's spike code for choir type of sounds, unlike the conclusion in section 5.9 in chapter 5.

6.10 Findings from PESQ on: Sounds from Nature

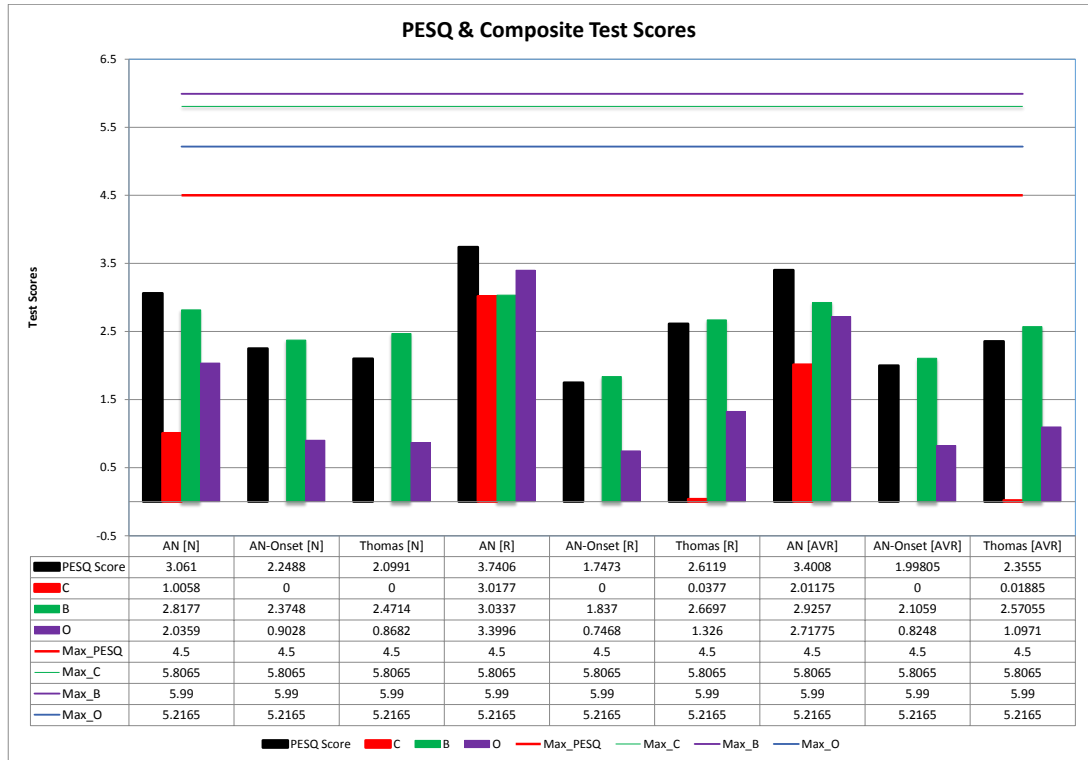


FIGURE 6.21: PESQ test scores for **Nature** type of sound: AN[N] represents the ‘cracking wood noise’ and AN[R] represents the crow song. AN[AVR] represents the average of those two. ‘Thomas[N]’ represents the PESQ scores from Koickal’s code for ‘cracking wood noise’ etc.

Two sounds have been analyzed here: Crow and cracking wood noise. Likewise the choir type of sounds, AN spike code has been very effective for coding both sounds, as they do not fall far apart from each other. However this is not true for the AN-Onset spike code and Koickal’s spike code, as their PESQ scores are quite far away from each other. Figure 6.21 explains this in detail.

Also, here the AN spike code has been better than the AN-Onset spike code, as the average PESQ score has been 3.4 for AN & 1.99 for AN-Onset. For the ‘cracking wood noise’, AN-Onset has been better than Koickal’s but for ‘crow’, Koickal code has been better than AN-Onset. The average scores suggest that Koickal’s spike code has been better than AN-Onset spike code. The number of spikes also follows the same trend discussed at section 5.9 in chapter 5.

6.11 Findings from PESQ on: Reverb Challenge

The ‘Reverb Challenge’ has been introduced to evaluate novel and established speech enhancement and recognition techniques in reverberant environments, provide an opportunity to the researchers in relevant fields to carry out comprehensive evaluation of their systems based on common data sets, and enable fair and reasonable comparison of different approaches, aiming at providing new insights into the problem and facilitating clear understanding of the state of the art [86].

Reverberation is basically echoing the original sounds by near or far surfaces. So, these spike generating coding techniques have been tested and evaluated in that reverberation environment to see which coding technique is more efficient for reverberated sounds. Matlab codes have been developed by using convolution function to generate the reverberated sounds (see MATLAB Codes in Appendix A). In this test, a countdown and clock tick-tock sound has been the original sounds. They are reverberated inside a small hall and a big hall. Obviously reverberation in small hall is shorter than the reverberation in big hall. The results are shown in the figure 6.22.

From the figure 6.22, we can say that clock in a big hall has been decoded best by AN spike code. Clock sound has also been coded better than the countdown sound in the reverberation environment.

Also, the reverberations in big hall have always been coded better by all three types of spike codes. In the figure 6.22, we can see the scores are always higher for the bigger reverberations.

The AN spike coding has been the best spike coder than AN-Onset spike coder, which has been better than the Koickal’s spike codes.

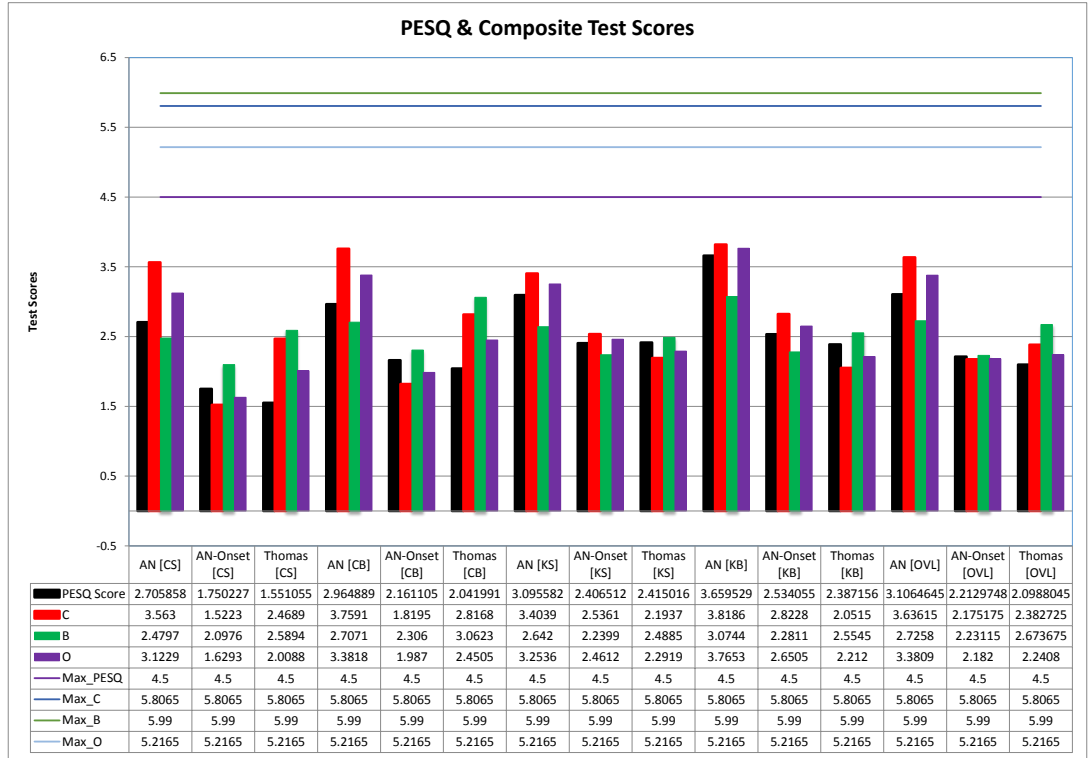


FIGURE 6.22: PESQ test scores for **Reverb Challenges**: AN[CS] represents the ‘countdown in a small hall’ and AN[CB] represents that countdown in a big hall. AN[KS] represents the ‘clock tick-tock in a small hall’ and AN[KB] represents that clock tick-tock in a big hall. AN[OVL] represents the average of those two. ‘Thomas[CS]’ represents the PESQ scores from Koickal’s code for ‘countdown in a small hall’ etc.

6.12 Findings from PESQ on: Noise

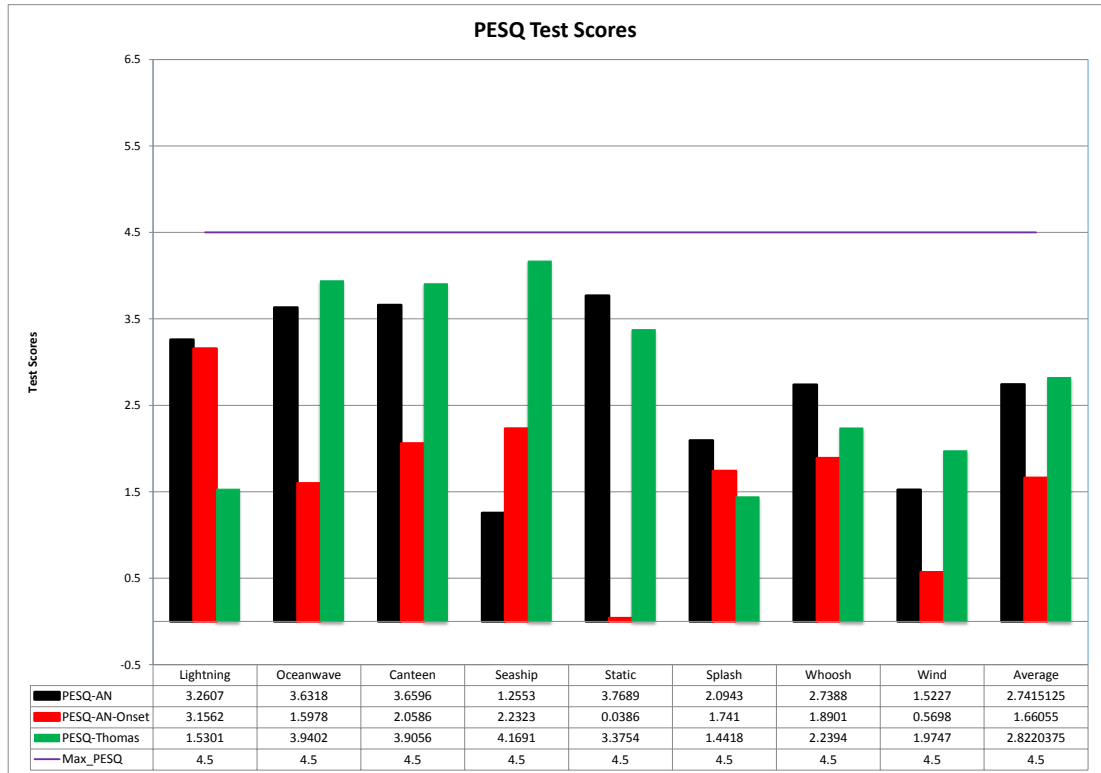


FIGURE 6.23: PESQ test scores for **Noises**: The PESQ scores are not very consistent with each other. ‘PESQ-Thomas’ represents the PESQ scores from Koickal’s code.

In this test, we have considered various types of noises including white noises, noise in a full canteen and wind noise. Figure 6.23 summarizes the PESQ test scores of all of those different types of noises and it shows that the values do deviate from each other quite a bit.

PESQ test scores for all noises			
Spike Coder	AN	AN-Onset	Koickal
Average	2.7415125	1.66055	2.8220375
Standard Deviation	0.940332395	0.90869901	1.071026835

TABLE 6.6: These are the average PESQ test scores for all the Noises with their standard deviations.

From table 6.6, we can say: the standard deviations are very high, where high standard deviation means that the scores have been deviated from each other too much. The average PESQ test score has been 2.74 for AN, 1.66 for AN-Onset and 2.82 for Koickal’s

spike code. So, the AN is the best as usual, but Koickal's spike code is better than AN-Onset spike code.

According to table 6.6, we can see that Koickal's spike code has the maximum standard deviation, which means that it varies too much from each other. So, Koickal's spike code can code some of the noises better than the other. This is quite true for AN & AN-Onset coding as well, as their standard deviation is as high as 0.94 and 0.91.

So, the spike coding technique is quite good for noises, however some type of noises like static and lightning can be coded much better than others, as in those types of sounds, there are sudden changes in spectral composition.

Number of spikes for noises				
Sound	Sound length	AN	AN_Onset	Koickal
Oceanwave	9.4624 sec	1268528	151653	550390
Canteen	8.2208 sec	1097078	131000	449957
Seaship	14.4969 sec	1911415	230140	798222
Static	2.5627 sec	282269	34602	178610

TABLE 6.7: **Number of Spikes :** Here the number of spikes for four noises have been mentioned along with the corresponding length in seconds. The numbers for each coding type are the total number of spikes used to decode the reconstructed sound in that coding technique. So, to decode the Oceanwave sound from AN_Onset spikes codes, 151653 spikes have been used.

From this table 6.7 we can say that for Koickal's spike coding technique, there are more spikes than AN_Onset spikes. But the quality of decoded sounds for Koickal's is much better than AN_Onset spikes. As AN is less lossy, AN spikes appear as too many and decodes the best quality of decoded sound. So, for the noises the number of spikes does really matter. Unlike the conclusion mentioned in section 5.9 in chapter 5, AN_Onset spikes cannot outperform the Koickal's spikes.

6.13 Findings from PESQ on: Signal-to-Noise Ratio (SNR)

SNR is another important issue considered here. Here we are interested to see which ratio is good or better for coding decoding sounds from its spike codes. The ratios which have been considered here are: 30dB, 20dB & 10dB. Three types of noises have been added at the background of a speech mixed by those ratios. They are: Canteen Noise, Seaship Noise and Static white noise. The MATLAB function doing this job has been mentioned here: (see MATLAB codes in Appendix A).

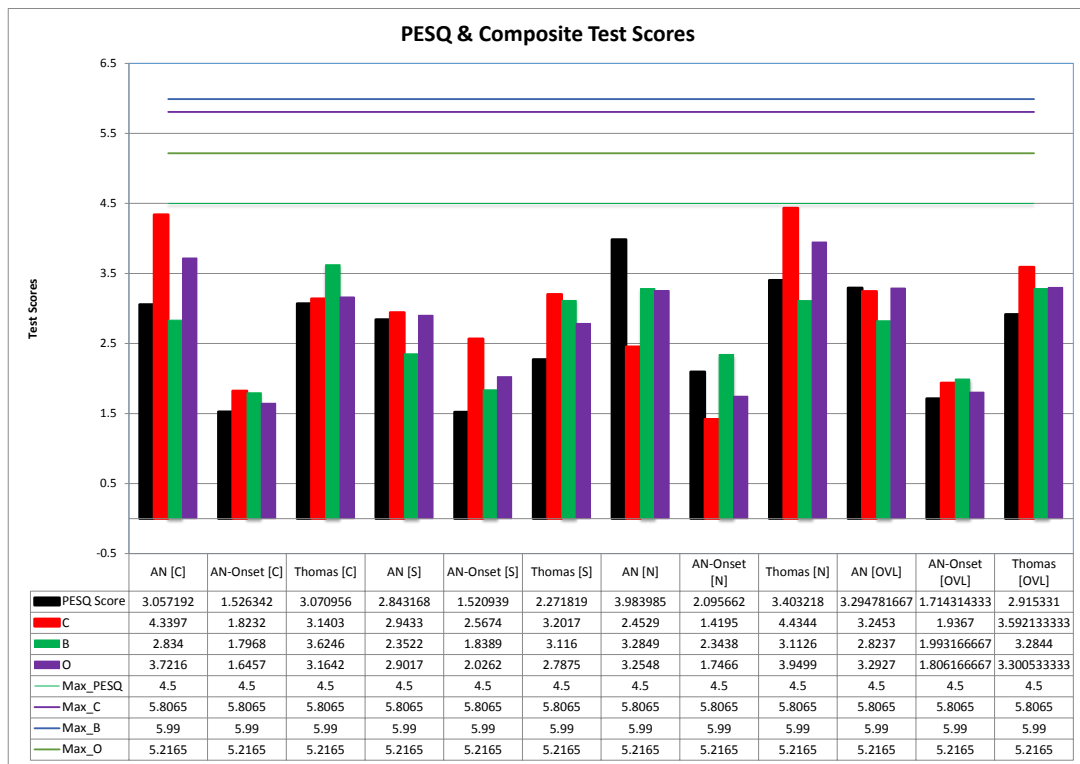


FIGURE 6.24: PESQ test scores for SNR 10dB. AN[C] represents the ‘canteen noise’, AN[S] represents the ‘sea-ship noise’ and AN[N] represents ‘static noise’. AN[OVL] represents the average of those three. ‘Thomas[C]’ represents the PESQ scores from Koickal’s code for ‘canteen noise’ etc.

For 10dB SNR, the figure 6.24 shows that for static noise, the AN coding has produced good quality of decoded sound. The AN-Onset coding and Koickal’s coding has followed the same pattern like decoding the static noise sound the best among the others. The canteen noise has been placed as second next to the static noise for all three spike coding techniques followed by the third sea-ship noise.

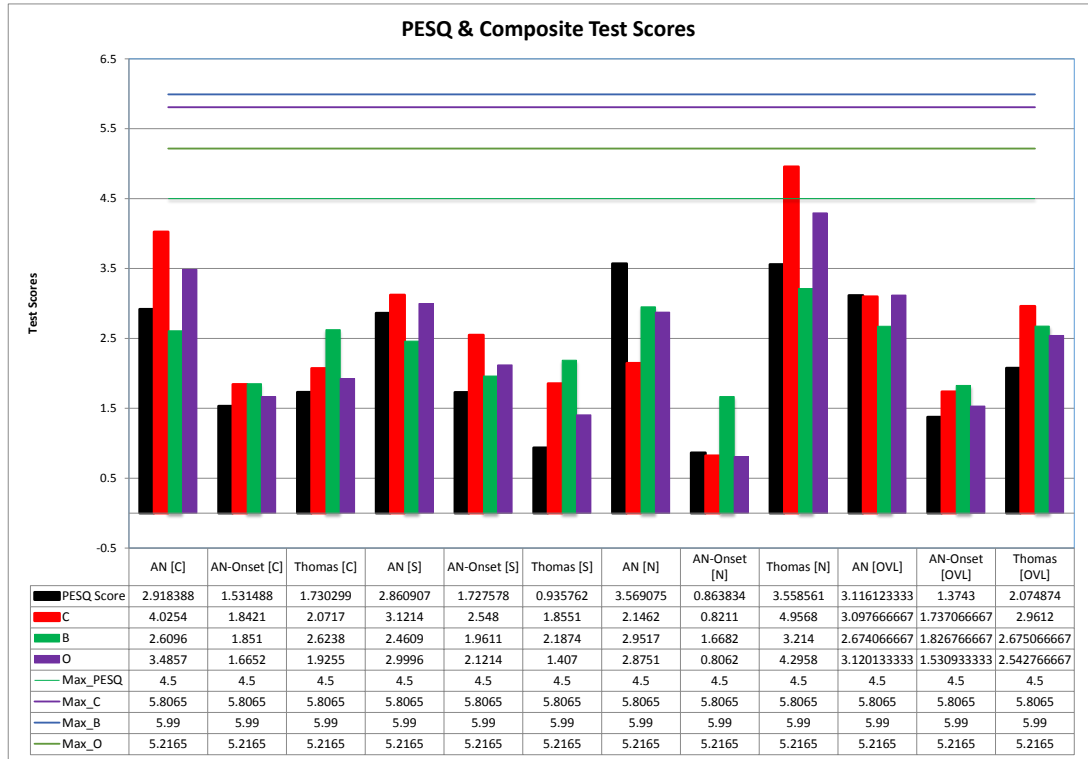


FIGURE 6.25: PESQ test scores for SNR 20dB. AN[C] represents the ‘canteen noise’, AN[S] represents the ‘sea-ship noise’ and AN[N] represents ‘static noise’. AN[OVL] represents the average of those three. ‘Thomas[C]’ represents the PESQ scores from Koickal’s code for ‘canteen noise’ etc.

So, for 10dB SNR the results follow the previous section’s ‘Noise’ type of sound’s results. The spike coding is quite good for static noise but the quality decreases when we consider the canteen and sea-ship background noises.

For 20dB SNR, canteen background noise one decreases its quality and so does the static background noise one. The Sea-ship background noise one actually holds its score value from 10dB to 20dB.

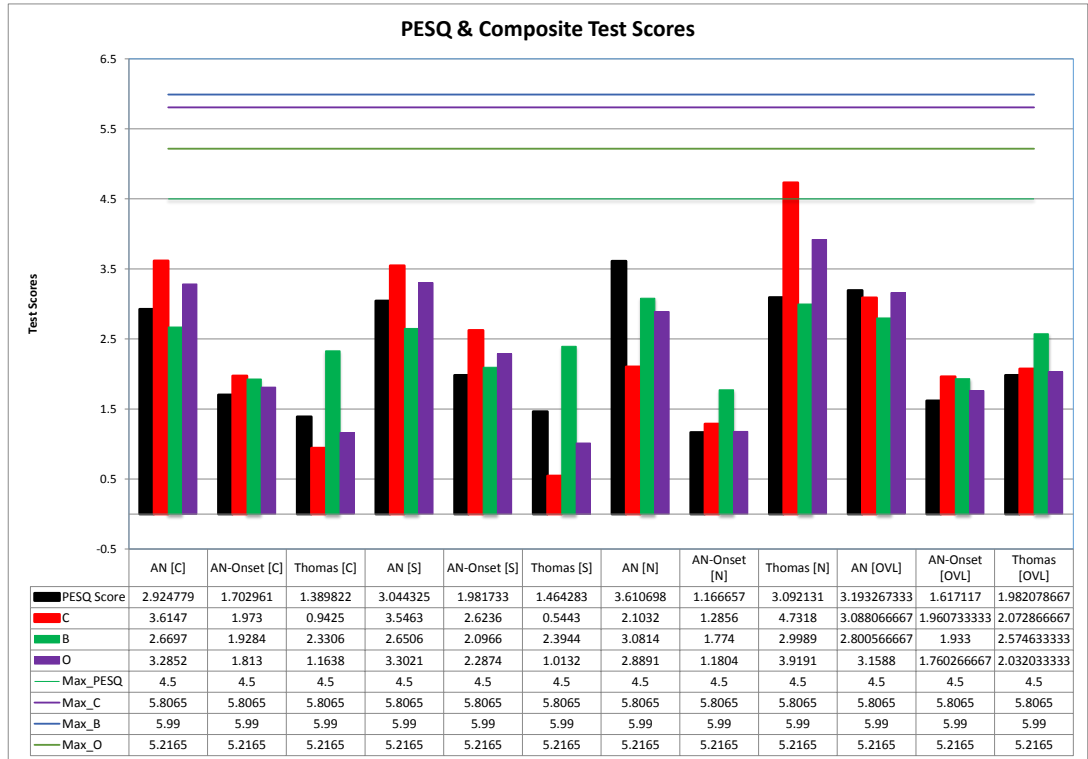


FIGURE 6.26: PESQ test scores for SNR 30dB. AN[C] represents the ‘canteen noise’, AN[S] represents the ‘sea-ship noise’ and AN[N] represents ‘static noise’. AN[OVL] represents the average of those three. ‘Thomas[C]’ represents the PESQ scores from Koickal’s code for ‘canteen noise’ etc.

For 30dB SNR, AN spike coding produces the best quality sounds for static background noises as the score has been 3.61. The Sea ship background noise one is little worse than the static noise having score 3.04 and the canteen one is the worst having the PESQ score 2.92. However AN-Onset coding produces the worst for static noise background decoded sound than other background noises. The Sea-ship background noise one is the best decoded sound among others for AN-Onset coding followed by the canteen background noise.

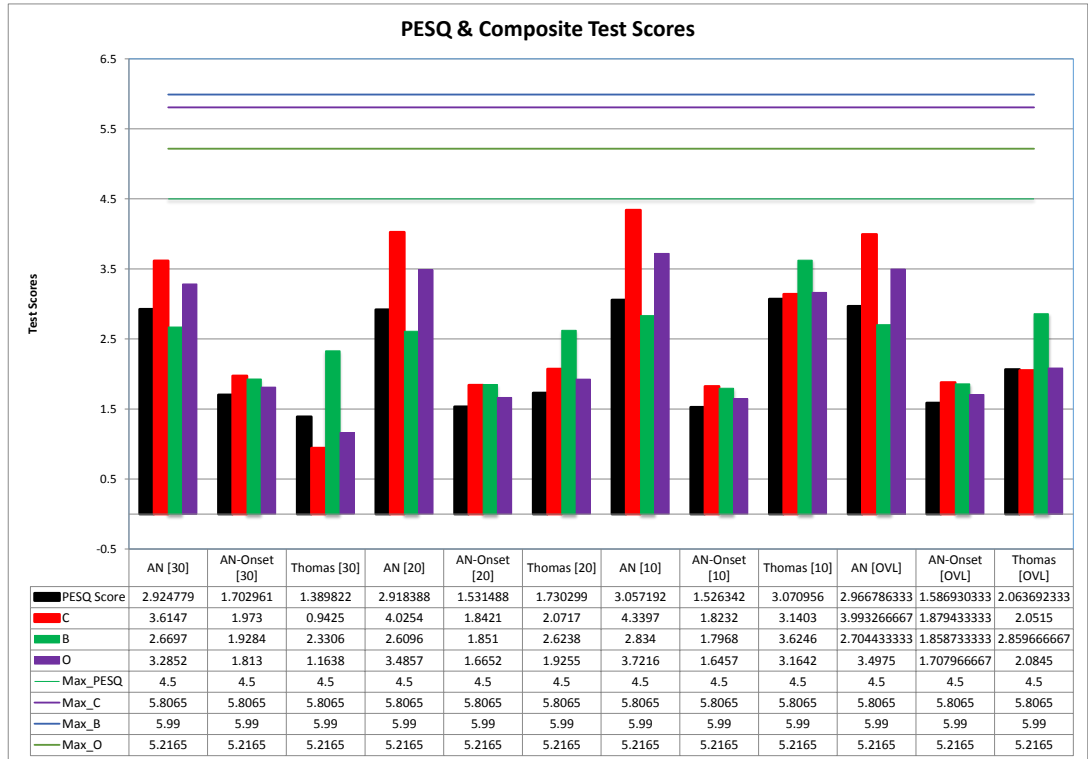


FIGURE 6.27: PESQ test scores for SNR (30, 20, 10dB) with background Canteen Noise. AN[30] represents the ‘30 db SNR’, AN[20] represents the ‘20 db SNR’ and AN[10] represents ‘10 db SNR’. AN[OVL] represents the average of those three. ‘Thomas[30]’ represents the PESQ scores from Koickal’s code for ‘30 db SNR’ etc.

For Canteen background noise, the test score has been quite consistent along with different SNR.

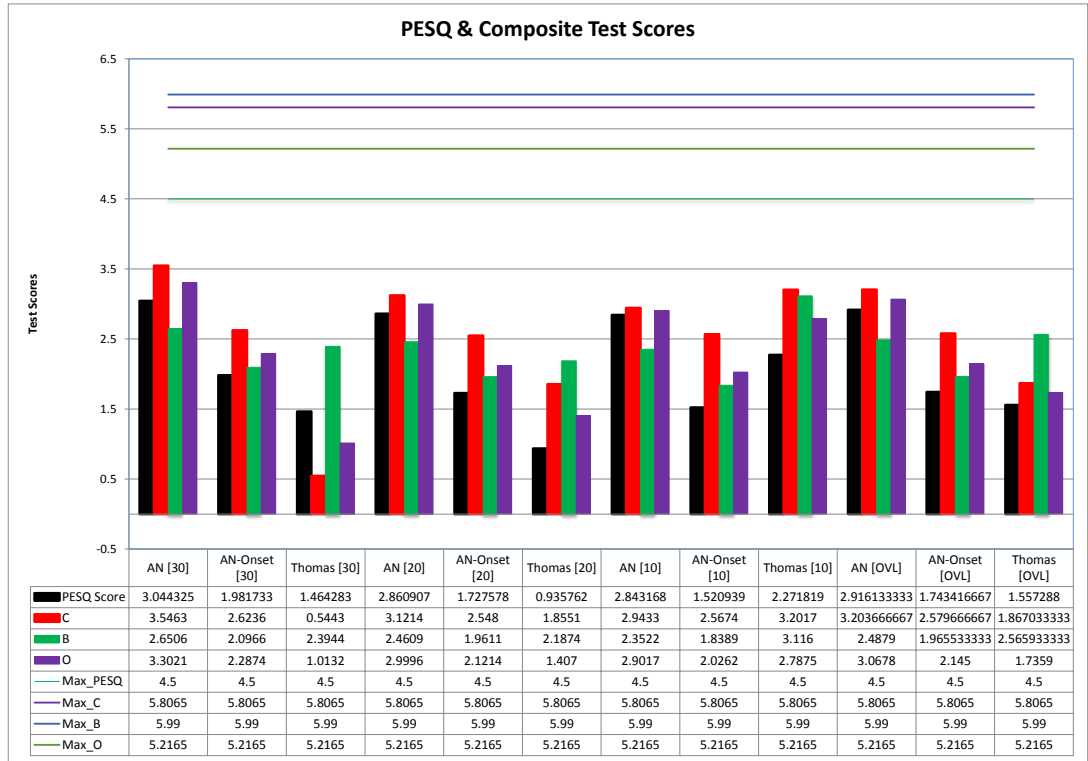


FIGURE 6.28: PESQ test scores for SNR (30, 20, 10dB) with background Sea-ship Noise. AN[30] represents the ‘30 db SNR’, AN[20] represents the ‘20 db SNR’ and AN[10] represents ‘10 db SNR’. AN[OVL] represents the average of those three. ‘Thomas[30]’ represents the PESQ scores from Koickal’s code for ‘30 db SNR’ etc.

For Sea-ship background noise, the test scores improved along with higher SNR for AN and AN-Onset coding technique, but not for Koickal’s spike coder. Koickal’s spike decoded sound significantly gets worsen for 20 dB however for other SNR it produces better quality of sounds.

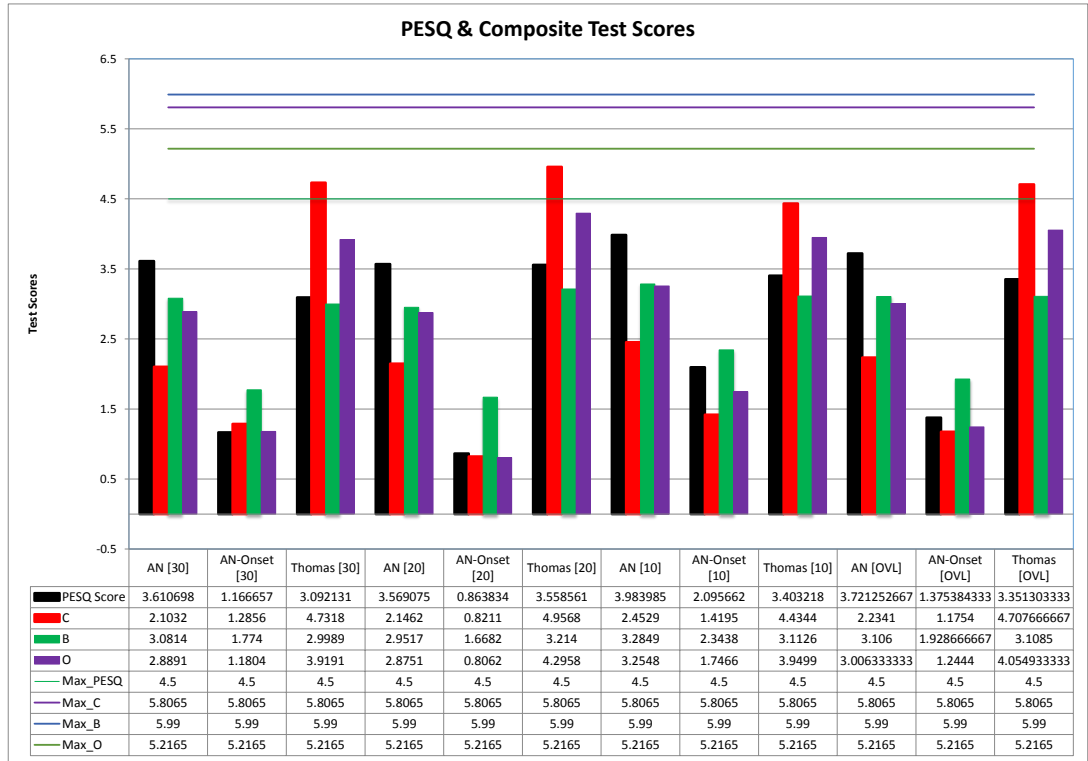


FIGURE 6.29: PESQ test scores for SNR (30, 20, 10dB) with background Static Noise. AN[30] represents the ‘30 db SNR’, AN[20] represents the ‘20 db SNR’ and AN[10] represents ‘10 db SNR’. AN[OVL] represents the average of those three. ‘Thomas[30]’ represents the PESQ scores from Koickal’s code for ‘30 db SNR’ etc.

For Static noise, the AN and AN-Onset spike coding produces the worst quality of sound for 20 dB and the best for 10 dB SNR. Koickal’s spike coding technique has been impressively good for decoding static background noise sounds as the average score for Koickal’s spike code has been 3.35. The reason Koickal’s spike coding technique has been so good for static noise is that it has been developed for artificial sound like static noise etc. Also, in static noise, the amplitude values are random so there is no pattern in the noise. AN and AN_Onset spike coding has been developed for recognizing pattern in the sound. So, for static noise, Koickal’s spike coding technique outperforms the AN and AN_Onset spike coding technique.

So, if the SNR is increased (from 10dB to 30dB) i.e. the intensity of noise is decreased, the quality of decoded sound gets better. This shows evidence that in general the spike coding technique is more accurate to code speech type of sound rather than noise. For different types of noises, however the spike coding technique works with different accuracy.

6.14 Findings from variations of PESQ and Composite Scores & Speech Recognition

The following figures explain the PESQ and composite test scores between various male and female utterances. 5 Male and 5 Female utterances have been coded and the decoded and the decoded sounds have been compared with the original sounds. Now, the comparison reveals the PESQ and composite scores for each male and female.

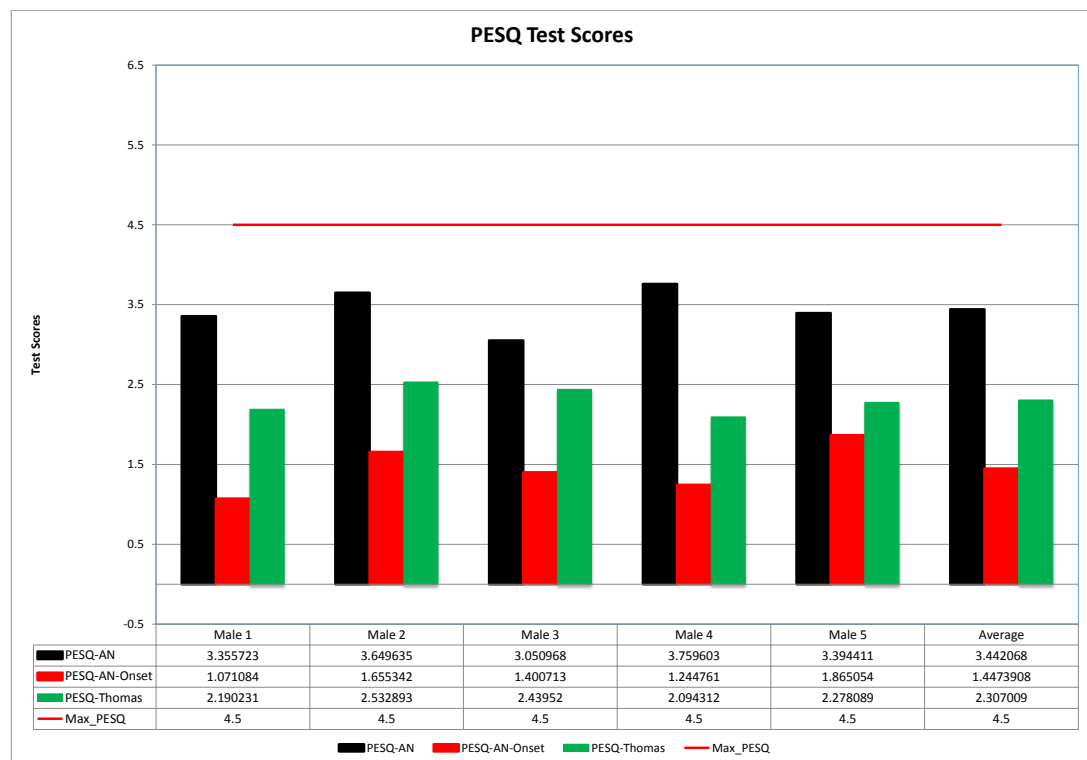


FIGURE 6.30: **PESQ test scores for male voices:** The standard deviation of PESQ scores among all those participants are: 0.247628106 for AN, 0.283736043 for AN-Onset & 0.160185036 for Koickal (PESQ-Thomas)

Figure 6.30 reveals the PESQ scores for 5 male speeches including the average PESQ among all those 5 male speech. Standard Deviation among those PESQ scores for AN, AN-Onset and Koickal's Decoding can provide a better understanding of how each male voice differs from each other. Standard deviation determines the amount of deviation of a set of scores. So, the coding technique which produces PESQ scores which deviates more for different types of sounds, will be more useful to recognize the difference between those sounds. The AN coding has a standard deviation of 0.2476, AN-Onset coding has a standard deviation of 0.2837 whereas Koickal's coding has a standard deviation of 0.16

of male voices. So, among all of these three coding, the AN-Onset provides the highest standard deviation which means that AN-Onset coding can be used to differentiate different male speakers. The AN-Coding is not far off from AN-Onset coding from its standard deviation either. It tells us that one sort of male speech can be coded better than other by AN or AN-Onset spike coding. The coding technique with highest PESQ score should be used for speech recognition as that technique can code the sound best.

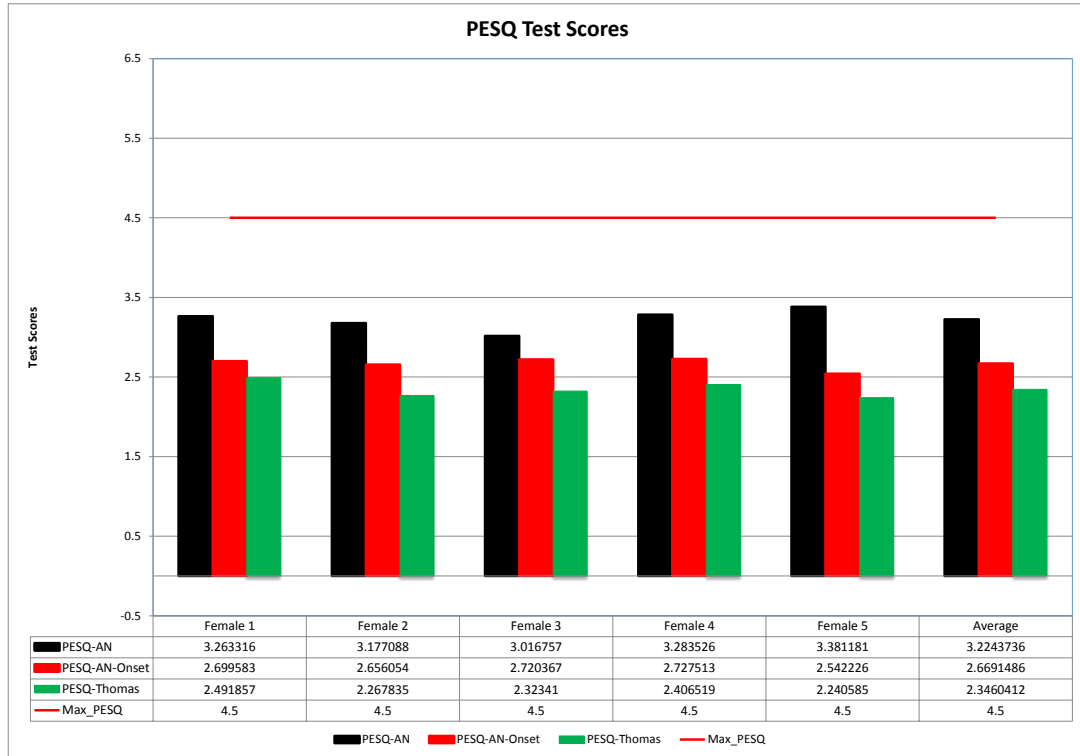


FIGURE 6.31: **PESQ test scores for female voices:** The standard deviation of PESQ scores among all those participants are: 0.122429292 for AN, 0.068171541 for AN-Onset & 0.092378789 for Koickal (PESQ-Thomas)

Figure 6.31 shows this for female speakers. It shows that the standard deviation is the highest for AN-coding which is 0.1224, unlike male speakers. So, for female speech, AN-coding can be used to differentiate them.

Now we are going to use the same approach to judge the quality of decoded sound on the signal distortion scores (C_{sig}). Figure 6.32 provides a details on the C_{sig} comparison for those male sounds. The AN-Onset coding provides the highest standard deviation among all the C_{sig} scores which is 0.4773 followed by Koickal code, which is 0.3493. AN-coding has the lowest standard deviation 0.1833.

So, again AN_Onset spikes can be used to recognize different speeches as it produces different quality of them for different male speakers.

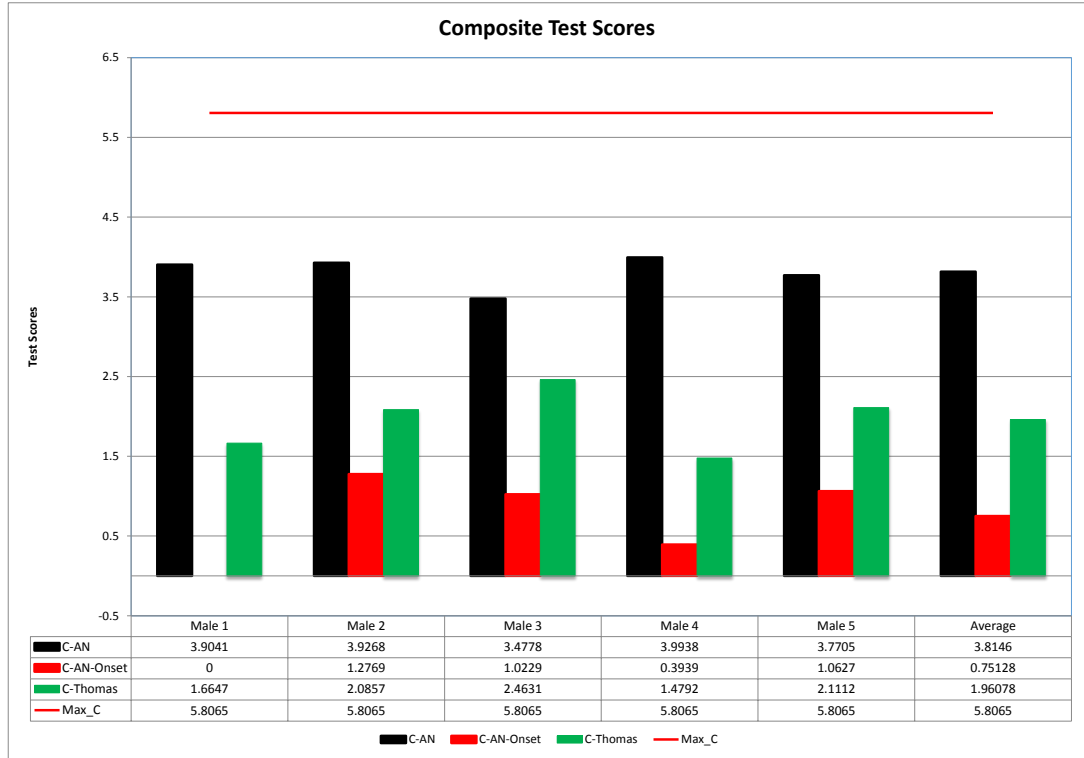


FIGURE 6.32: **C test scores for male voices:** The standard deviation of C_{sig} scores among all those participants are: 0.183352546 for AN, 0.477321344 for AN-Onset & 0.349324843 for Koickal (C-Thomas)

From this figure 6.32, we can see that AN coding technique has produced the decoded signals with least signal distortion. The other two - AN_Onset coding and Koickal's coding technique produce C_{sig} scores which are far away from the AN coding's C_{sig} scores. The standard deviation is highest for AN_Onset coding technique for male speech.

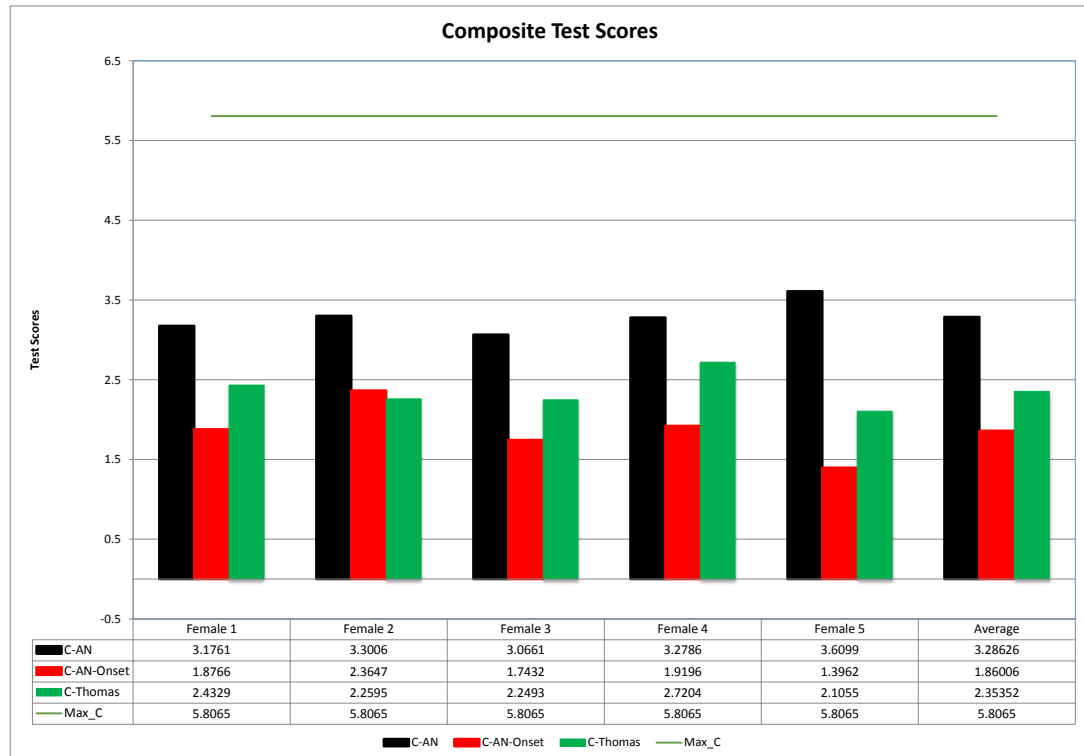


FIGURE 6.33: **C test scores for female voices:** The standard deviation of C_{sig} scores among all those participants are: 0.181996051 for AN, 0.312186634 for AN-Onset & 0.210768882 for Koickal (C-Thomas)

The results are similar like male speech for female speech as well as AN coding technique outperforms the other coding techniques by the quality of decoded sounds for signal distortion.

For both male and female speech, the highest standard deviation has been achieved for AN-Onset coding as well, according to figure 6.32 and 6.33.

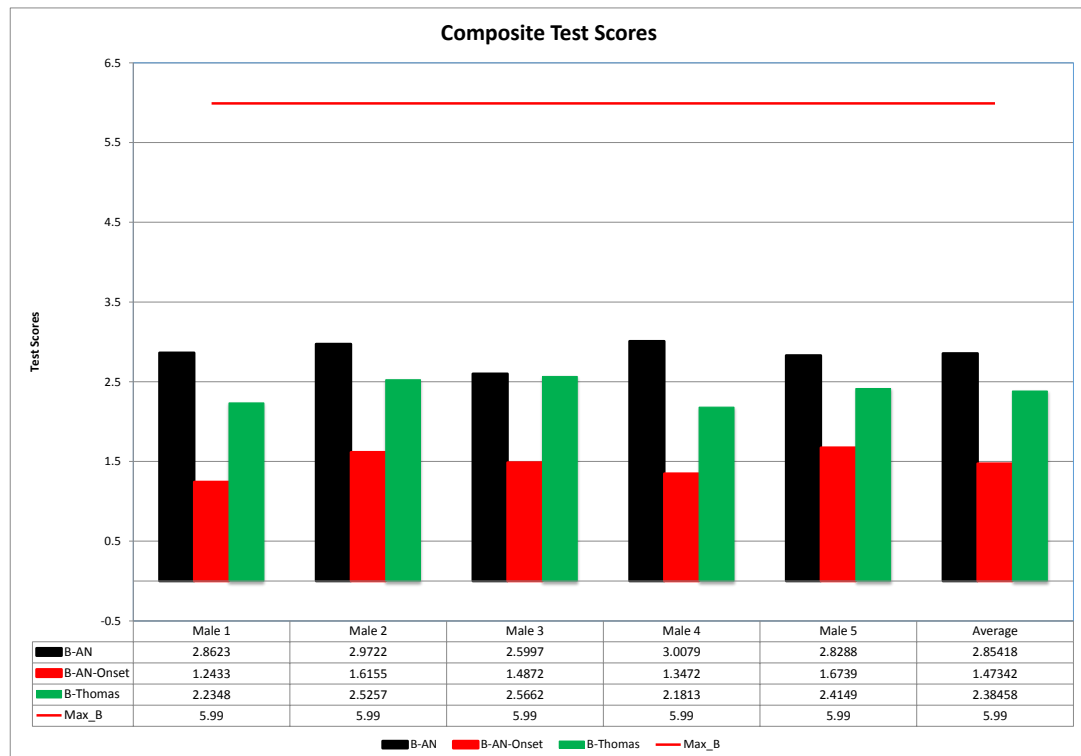


FIGURE 6.34: **B test scores for male voices:** The standard deviation of C_{bak} scores among all those participants are: 0.143547001 for AN, 0.160907146 for AN-Onset & 0.15334718 for Koickal (B-Thomas)

Like signal distortion, AN coding technique produces the decoded signals which have least background noise distortion for male speech in figure 6.34. AN-Onset coding produces the highest standard deviation among other coding techniques.

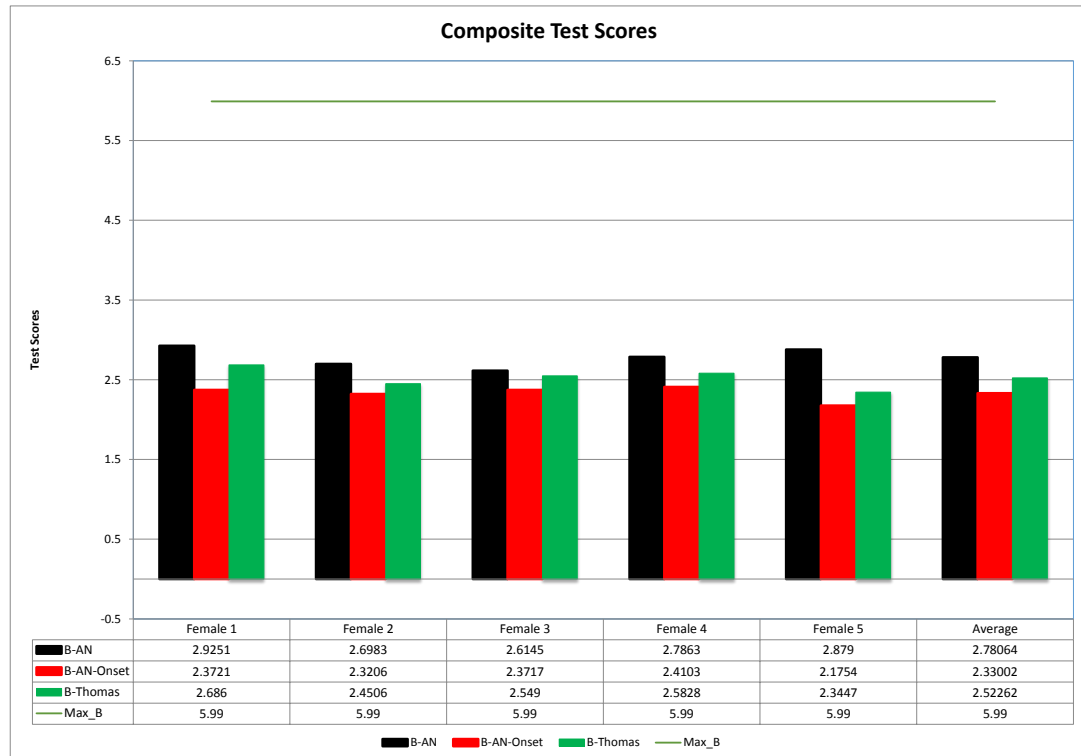


FIGURE 6.35: **B test scores for female voices:** The standard deviation of C_{bak} scores among all those participants are: 0.113980518 for AN, 0.082400131 for AN-Onset & 0.116492891 for Koickal (B-Thomas)

For female speech, background noise distortion is not much different between the decoded sounds from AN and AN_onset coding unlike male speech. AN coding has slightly higher C_{bak} scores than AN_Onset coding. AN-Onset coding provides the highest standard deviation among other two coding techniques, as figure 6.35 shows.

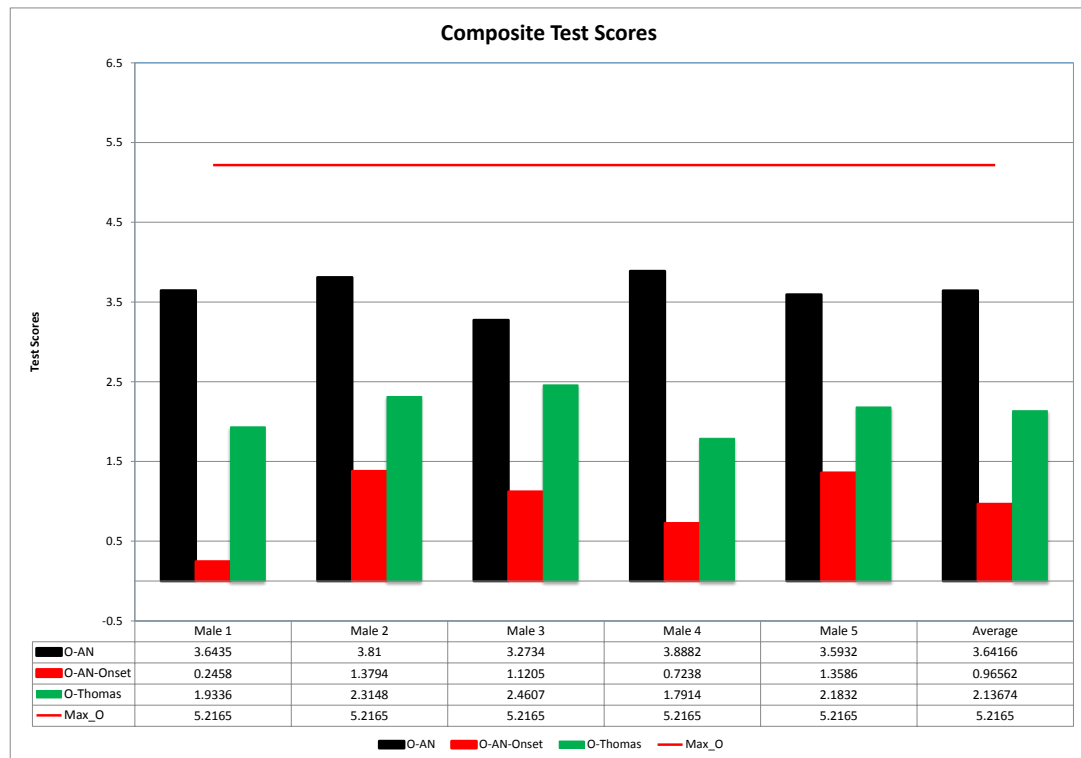


FIGURE 6.36: **O test scores for male voices:** The standard deviation of C_{ovl} scores among all those participants are: 0.213113825 for AN, 0.430407418 for AN-Onset & 0.244679199 for Koickal (O-Thomas)

Overall for male speech, AN spike coding produces always better quality of decoded sounds than other two coding techniques and AN_onset coding always has the highest standard deviation among the other two.

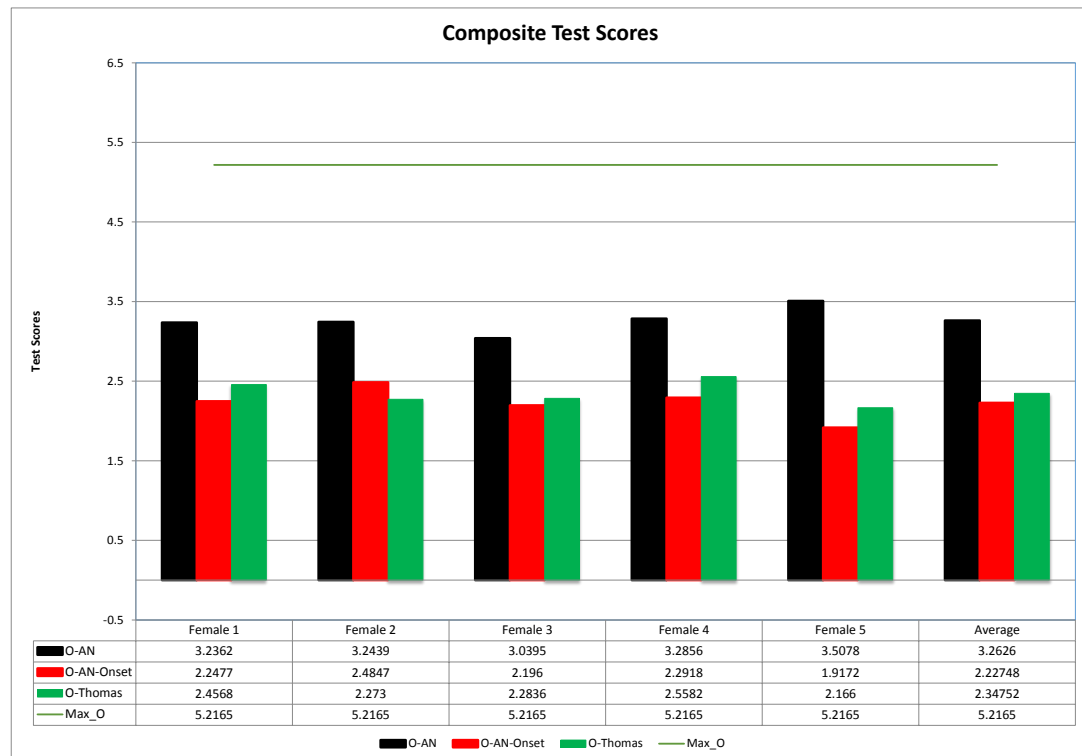


FIGURE 6.37: **O test scores for female voices:** The standard deviation of C_{ovl} scores among all those participants are: 0.149313295 for AN, 0.183288138 for AN-Onset & 0.140653253 for Koickal (O-Thomas)

Overall for female speech, AN spike coding produces always better quality of decoded sounds than other two coding techniques. The overall composite scores differ at its maximum for AN-Onset coding for male and female speakers as shown in figure 6.36 and 6.37.

The coding which can code sound better than other coding techniques should be used for speech recognition, as it can code more information than others. As, AN coding provides highest PESQ scores, although with some variations, it should be used for speech recognition.

6.15 Findings from PESQ on: Male Voice & Female Voice

Figure 6.30 and 6.31 provides the average PESQ test score for AN spike coding of 5 males and 5 females. They are 3.2244 for females and 3.4421 for males. So, the male PESQ score is 6% higher than female. However for male speech the standard deviation is 0.25 which is higher than female speech standard deviation 0.12.

So, this provides evidence that the spike coding technique works better for male speech rather than female speech. The reason behind that is male speech has a longer low frequency contents than female.

6.16 Possible Future Sound Testing and Pros and Cons

The objective sound testing which has been applied here is PESQ and composite test scores [65] and [66]. Quite a large number of sample sounds have been tested and reported by their test scores. However there are other tests which can be done to compare these sounds and the coding technique efficiency. One of the other testing can be done by the Objective Matrices as suggested by Salford University Acoustics research group [87]. Here they have advised a closed systems which have advantages of ease of use but less flexible ([88]).

National Acoustic Laboratories of Australia uses objective sound testing as well [89]. They use Electro-physiological testing for both adults and infants. It requires minimal co-operation from the participants.

We have used PESQ and composite test scores as they are very reliable, easy to use and most popular ([65] and [66]). However, other objective sound testing can be used on the sounds which we have used in our testing and the results can be obtained and compared with PESQ composite testing. They can provide a wider knowledge about the quality of spike coding.

Chapter 7

Conclusion

7.1 Summary of the Research Work

This research has focused on sound coding based on event based (spike) representation. We have mainly considered three spike coding techniques. Two of them are biologically inspired and the third one is hardware oriented. The reason for decoding the sounds from the spikes is to investigate the efficiency, quality and practicality of these spike coding techniques. The decoding of spike codes also provides the decoded sounds which can be compared with each other to find out which spike coding is better.

Also, in this research the size of the files produced has been considered when the spike codes are generated. That provides a way of comparing the coding efficiency of those spike codes.

The decoding technique of AN spike representation has been described in chapter 3. It has also been explained that even although our brain does not reconstruct the sound from the auditory nerve like fiber spikes, the decoding provides a useful comparison between the original and decoded sound to identify the effectiveness and genuineness of those spike coding techniques. There are many issues which are involved in this decoding technique as described in chapter 3 like: delay vectors, proper tuning of GMF, number of channels and sensitivity levels and others. The processing time of coding and decoding spikes has also been reduced by inventing a new spike database.

The onset spikes generation technique has been investigated by its parameter values as discussed at the beginning of chapter 4: depressing synapse parameters, onset cell weight and number of AN fibers connected to each side of cells. The final tuned parameters generate onset spikes at the beginning of each increase of energy in a sound signal. To test the effectiveness of those parameter, a amplitude modulated sound has been generated modulated at 120 Hz with a carrier frequency of 1500 Hz. 120 Hz is almost the fundamental frequency of a male voice speech and 1500 Hz covers the middle section of the GMF channels. That's why these two frequency values are chosen for tuning the onset parameters so that onset spikes can also be used as AM detector. Once the parameters generate onset spikes, the decoding method has been developed by combining AN spikes, AN-Onset (frequent version of onset spikes) & Onset (original onset) spikes. The algorithm has been documented in chapter 4.

Koickal's spike coding technique is described in chapter 4. So, all three different spike codes have been finely tuned and then decoded from its coded states. Then it was the time for testing to evaluate the efficiency and accuracy of those spike codes by comparing decoded and original sounds. Firstly a subjective test has been carried out. 21 volunteers participated with different age, gender and hearing ability in that sound test of 20 questions. The answers are hypothesized binomially to provide the statistical evidence behind any conclusion. This subjective testing has been described in chapter 5. Then an objective testing has been carried out as mentioned in chapter 6. PESQ and composite tests provides some scores to find out how similar one sound is to another. The same sounds, which were used in subjective testing, have been evaluated in the objective testing and the results obtained were the same. As this testing takes less time and easy to carry out, most of the factors affecting the reconstruction work has been tested. The issues were: the number of channels and sensitivity levels, onset parameters, different types of sounds (musical notes, voice, choirs and others).

7.1.1 Technique 1: AN Spike Coding

The spike codes are generated and saved according to each channel (or frequency) and sensitivity levels. The decoding technique has been described in chapter 2. However there are a few issues rose that affected the decoding technique significantly. One of them was the effect of delay vectors. The AN spikes are generated from the bandpassed signals.

The filterbank, which was used to generate AN spikes, delays each channel frequency (plotted in figure 3.2). The sounds were generated with and without compensating for the delay vectors and they were plotted with the original signal to describe the effect in detail.

Decoding from AN spikes has been accomplished by using different numbers of channels and sensitivity levels. Originally 50 channels and 16 sensitivity levels were used to code and decode the AN spikes. However that combination has been changed and the sounds have been generated for different number of channels and sensitivity levels. Those sounds were compared to the original sound through the objective testing: PESQ and composite testing. The test scores have been obtained and then conclusions have been made based on those scores and the amount of kilobytes it requires to store the corresponding spikes. The final conclusion was to use 50 channels and 16 sensitivity levels.

Auditory nerves cannot fire more than 200 spikes in a second. So, this MSR technique has been applied to code the spikes and then decoded to analyze the quality of that decoded signal. In chapter 2, it has been explained by the mathematical equations and graphs. Figures 3.15 and 3.18 explain that there are some concentrated energy appears for each frequency channels. Figure 3.13 explains that we lose information at the high frequency where the spikes appear very frequently. To examine the effect of jittering the spikes and to distribute the extra energy contents in high frequency channels, the spikes have been jittered by its time-domain appearances by 1% and 10%. This successfully distributes the energy contents but cannot provide a good regenerated sound as the spikes are scattered randomly and real information is lost in this coding technique.

The AN spike codes have been applied to noisy signals as well. This has been done by introducing the Signal-To-Noise ratio (SNR). The mixture has been chosen as 30 dB, 20 dB and 10 dB. Then it has been tested objectively by PESQ by mixing three different types of noise in the background of speech. If the SNR is increased (i.e. from 10 dB to 30 dB), then the quality of decoded sound is decreased. The spike coding techniques work better for the static background noise where no pattern can be found.

Also, when regenerating the sounds, the sine waves were generated according to each spikes and they were ramped and smoothed to cut down any significant noise in the decoded sound. And at the end, the AN spike generation technique has been amended so that the spike-generation time can be minimized.

7.1.2 Technique 2 & 3: Onset Spike Coding & Koickal's Spike Coding

The onset spike generation technique is complex with many parameter values. As explained in chapter 4 at section 4.3, the values of α , β and g have been tuned to a reasonably good combination which provides the onsets at each peak of sound signal amplitude. At the objective testing section, several combinations have been used to find out which combination might work best for speech.

7.1.3 Testing

Testing was crucial in this research to examine the quality of many decoded signals. Firstly, a subjective testing has been commenced where 21 people participated with different age ranges and genders. Two sounds were put in a website where the participants can play them and choose the better sound. Their answers were recorded and a Binomial test has been carried out to find statistical evidences to choose one sound over the other. Chapter 5 explains this subjective test in detail.

Then objective testing PESQ and composite tests have been carried out over many different sounds. This test method is particularly helpful to understand how much signal distortion or background noise is present in the decoded signal compared to the original signal. Chapter 6 explains these objective tests in details with corresponding findings.

7.2 Empirical Findings

Both subjective and objective testing provides the empirical findings in this research. In table 5.41, each spike coding technique has been compared with other and the answer has been provided as better, same or worse. For four different types of sounds, it has been found that AN is the best among three spike coding techniques, as it provides the most similar decoded sound with the original. The AN-Onset spike coding provides similar decoded sound with original, but not as same as the AN spike's decoded sound. So, generally AN-Onset spike coding is worse than AN spike coding. Koickal's spike coding is worse or similar to the AN-Onset spike coding.

The objective testing provides a few other empirical findings as well. It agrees with the findings from the subjective testing which is AN spike coding is the better than AN_Onset spike coding which is better than Koickal's spike coding. The number of channels to be used in AN and AN_Onset spike coding is 50 channels and 16 sensitivity levels ultimately. The values of onset generating parameters, discussed in section 4.3, have been tuned in respect of the quality of decoding sounds. The final values of those parameters are $\alpha = 500$, $\beta = 25$ and $g = 1100$. The resolution bits for Koickal's spike coding technique should be set to '7'. For Musical notes, AN spike coding is better than AN_Onset coding and AN_Onset coding is better than Koickal's coding. Interestingly, Koickal's spike coding is better than AN_Onset spike coding for choir type of sounds. For other natural sounds and sounds with reverberation, AN spike coding is always better than AN_Onset spike coding and AN_Onset spike coding better than Koickal's spike coding.

The noise and SNR have also been considered as an environment, to see how well the spike coding techniques are or which spike coding is better than others. It has been found that for random noises like static white noises, the spike codes decode better than other noise with pattern in it like: canteen noises or sea-ship noises.

There was a possibility of using the spike codes for speech recognition, so five different male and female voices have been tested under three spike codes and it has been found that the male speech can be better represented by AN spike code than female speech, so it can be used for speech recognition. However, female speeches vary more than male speeches for AN spike coding.

7.3 Thesis Contribution & Limitations

In chapter 1 (introduction), there are research questions which were raised as a part of the research. The answers of them are discussed below -

First

The three spike codings we have considered are: AN spike coding, Onset spike coding and Koickal's spike coding. As the spike codes represent the sound, it will be possible to decode the sound from its spike coded

state. So, how can an algorithm be presented for reconstructing sound from its spike codes and is there any issues which to be considered in case of decoding?

The algorithms for decoding spikes from its coded state has been developed and described in chapter 3 and 4. Chapter 3 explains the AN spike decoding algorithm and the issues which were raised in decoding and the way those were handled. Chapter 4 discusses about the onset spike decoding from its coded state and the other issues which were handled properly.

Second

How can the AN and Onset spike coding techniques be more efficient to interpret a sound better and generate the spikes quicker?

The AN and Onset spike generation system has been improved so that the spike can be generated quicker and can accurately represent the sounds. It has also been improved so that the decoding process becomes much quicker. The GMF has been tuned properly so that the low pass-filtered sounds can be coded properly. Also, the original AN spike codes generated the AN spikes independently in each sensitivity level. The spike generation technique has been improved where the spikes are assigned by both its channel and sensitivity levels. Because of this improvement, the spikes can be decoded much quicker as well. This has been demonstrated in chapter 3. Figure 3.28 and 3.29 show how long it used to take to code and decode a particular sound. Figure 3.30 and 3.31 show that the much less time has been consumed to code and decode that particular sound under the improved technique.

The Onset parameters have been tuned as well so that the onset spikes can be generated at the beginning of every amplitude rise. This work has been demonstrated at the beginning of chapter 4.

Third

Sound coding is a very common and useful technique today and there are various lossy sound coding techniques like MP3, MP4, WAV etc. Spike coding is another type of lossy sound coding. In general, MP3 and MP4 are quite good at coding sound as the decoded sounds from them are very clear and of good quality. So, the research question has been raised that which spike coding technique is better to represent a

sound?

Among three spike coding techniques, AN spike coding provides the best output quality as it is less lossy than others. Between AN_Onset spike code and Koickal's spike code, both are good for certain types of sounds. In the subjective testing, we have considered four different types of sounds and in conclusion, we have mentioned that original sound is always better than spike decoded sounds, without doubt. So, spike representation cannot produce decoded sound as good as the original sound. According to the subjective test among the decoded sounds, AN-decoded sound is better than AN-Onset decoded sound & AN-Onset decoded sound is better than or same as Koickal's decoded sound.

Also, the number of spikes is an important factor to determine which spike coding is more effective and efficient than the other. The section 5.9 explains the number of spikes generated for each type of spike coding technique. In the table 5.42, four different types of sounds have been decoded by those three different spike coding technique and the corresponding number of spikes have been mentioned. The conclusion has been drawn that AN spikes are the best lossy coding technique. It generates better quality of decoded sound than other two. AN_Onset generates least number of spikes and generates better quality of decoded sound than Koickal's spike coding technique. Koickal's spikes are more in numbers than AN_Onset but worse in quality than AN and AN_Onset spikes. So, AN_Onset spike coding technique outperforms Koickal's spike coding technique. However there are a few exceptions which are mentioned next.

Fourth

There are various types of sounds like: male speech, female speech, choir, speech with background noise as different level of intensity. Which type of sound can be represented the best by a spike code technique?

The answer of this question has been achieved by running the Objective test, which is easier and quicker to carry out than subjective test. The results from objective testing are generally similar to subjective testing except that in some cases Koickal's decoded sound stands out better than the AN-Onset decoded sound. AN code can decode Male Voice type the best and higher frequency String type sound the worst according to figure 6.3. AN_Onset code can decode lower frequency String type sounds the best and Percussion type sounds the worst. Koickal's spike

code can decode the Male voice the best and higher frequency string type sound the worst according to figure 6.4.

More channels and sensitivity levels means better quality of decoded sound but also means more bytes required to store spikes. So, the threshold value - 50 channels and 16 sensitivity levels have been chosen as final. Likewise the resolution bits parameter in Koickal's spike code has been set to '7'. Also, for the lower frequency type musical notes AN-Onset coding technique works much better and almost as good as AN coding (see figure 6.16). The choir and some other natural sounds can be better represented by Koickal's spike coding technique rather than AN-Onset spike code. For Noise and SNR, the spike codes tend to work well for static noises and the similar types of noises. But the noises where there is a pattern like sea-ship noise or canteen noises are not decoded as well as static noises. For high intensity background noises (10dB SNR), the same results follow but it changes when the background noise becomes lighter (30dB SNR).

The average PESQ scores for both male and female speakers show that for AN spike coding, the male speech has been coded better than female speech.

As mentioned in the previous section and in section 5.9, AN_onset coding is better than Koickal's technique. But for choir type of sounds and for noises, Kiockal's technique generates better quality of decoded sound than AN_Onset technique keeping the number of spikes higher than AN_Onset spikes. So, we can conclude that AN_Onset spike coding technique is better than Koickal's spike coding technique most of the times but not all the time.

Fifth

Can the spike code be possibly used in speech recognition?

The spike coding techniques generate the decoded sounds which are clear and easy to understand, although the quality of them has been decreased. It is clear that the codes keep the information required for speech recognition, as shown by the fact that the decoded sound is easily recognizable. So, the spike coding techniques can be used for speech recognition.

Despite having these empirical findings, there are some limitations in the results and throughout the thesis. They are mentioned as follows:

1. Only the types of sounds, mentioned earlier in the thesis, have been used and tested. In nature, there are many more different types of sounds.
2. The onset spike generating system has been optimized by tuning the parameters. However that tuning was appropriate for only for the testings carried out in this thesis and they could be be improved. In another words, those parameters are not the best-tuned for all speeches so the decoded sound is not the best possible quality.
3. In the subjective testing, only 21 participants volunteered. So, the decisions for subjective testing have been made based on those 21 answers. The number of participants can be greatly increased. Also, only 20 questions were asked in that subjective testing, which took about 15 minutes on average to complete the whole test. The number of questions can also be increased and lengthier tests with appropriate breaks and refreshments can possibly be carried out.
4. In the objective testing, both PESQ and composite test scores have been received. However PESQ test scores have mostly been emphasized. The signal distortion, background noise and overall test scores can be used to achieve more detailed results.

7.4 Future Works & Final Remarks

This research work can be extended towards many future possible research projects.

1. **Improving Our Spike Based Techniques :** The spike coding techniques can be improved in the following ways -
 - (a) Because the AN_Onset techniques aims to discover AM in medium and high frequency bandpassed channels, it should be adjusted to reach for AM at appropriate frequencies. In chapter 4, it was adjusted for AM at about 120 Hz, which is appropriate for male voiced speech. For female and child speakers, where the fundamental frequency in voiced speech is higher, the AN_Onset technique should be adjusted to be sensitive to AM at a higher frequency.

- (b) The systems have been tested both subjectively and objectively. However more testing, particularly subjective testing could be carried out. This does however requires additional subjects to be available. For the objective testing, a wider range of sounds could be used and a more sophisticated approach to the AN_Onset technique can be adopted as mentioned in the previous paragraph.

2. **Applications :** This spike based systems can be applied to many ways as mentioned below -

- (a) **Cochlear Implant Processing :** The work here suggests that all the information present in this simplified versions of the auditory nerve can be sufficient for resynthesizing sounds effectively. This suggests that using pre-processing techniques similar to those described in chapters 3 and 4 may provide a good method for creating a spike-based representation that would be useful in a Cochlear Implant ([90]).
- (b) **Hearing Aids :** The effectiveness of the onset and AN_Onset like techniques could be a useful addition to the set of techniques used in preprocessing for hearing aids.
- (c) **Neuromorphic Systems :** This type of coding should be applicable in auditory neuromorphic systems. There are already many examples of the filterbank and spike generations being implemented in Neuromorphic systems ('Silicon Cochlea Building Blocks', chapter 9 in [91]). However this work suggests that these should be extended to include onset detection, both of the Onset and AN_Onset types. Neuromorphic systems are used in various circumstances like sensory systems, modeling neural systems, analog signal processing. As mentioned in [51], using spike based coding system a neuromorphic microphone can also be developed where sound transduction has a neuromorphic component.
- (d) **Sound Recognition Systems :** The effectiveness of the resynthesis in the systems with relatively low number of spikes suggests that something useful is being conserved in the lossy encoding. This, in turn, suggests that some of the features (encoded by the spikes) might be suitable inputs for a feature-based sound recognition system. This could provide a biologically inspired

move away from MFCC-type (Mel-frequency cepstral coefficients) techniques, where the frequency bands are equally spaced to approximate human auditory system for audio comparison.

- (e) **In Neuroeconomics :** To make decisions, our brain's nerve cells send signal from one to another. These signals can be called spikes (according to [92]). Our spike based system is based on the auditory nerve responses, however it can be extended to develop a model for nerve responses to make decisions ([93] and [94]).

3. **Remarks :** Our brain does not reconstruct the sound, it interprets it. But by decoding sound from its coding, we can measure the effectiveness and efficiency of a coding technique. This is true for any coding technique, whether spike-based or not.

Appendix A

MATLAB Codes for Reconstruction of Sounds from AN and AN_Onset spikes & Other Issues

The MATLAB codes to regenerate sounds from its spikes are available in web addresses as mentioned below -

Reconstruction of sounds from AN spikes (MATLAB code) :

In the attached DVD: `LIBRARY\MP_Lib\AN Sound Reconstruction\AN_Reconstruct.m`

In the attached DVD: `LIBRARY\MP_Lib\AN Sound Reconstruction`

Reconstruction of Sounds from AN_Onset spikes (MATLAB Code) :

In the attached DVD: LIBRARY\MP_Lib\AN & Onset Sound Reconstruction\
AN_and_Onset_Reconstruct.m

In the attached DVD: LIBRARY\MP_Lib\AN & Onset Sound Reconstruction\
AN_and_Onset_Reconstruct_GenerateSignal.m

Reconstruction of Sounds from Koickal's Spikes (MATLAB Code) :

In the attached DVD: LIBRARY\MP_Lib\Thomas Code toWork\
Thomas_code_Work.m

Reverberation of Sounds (MATLAB Code) :

In the attached DVD: LIBRARY\MP_Lib\Reverberation

Generating Signal-to-Noise ratio involving two sounds :

In the attached DVD: LIBRARY\MP_Lib\AN Spike Construction\
AN_Construct_SignalNoise.m

Appendix B

Reducing Spike Generation Processing Time (MATLAB Code)

```
function [zc_struct,assigned_Spikes,bmSig,threshold_level] = ...
    AN_SpikeGen_Mono_MJN_inNewSpikeForm_EditMP(bmSig,n_channels,cochCFs,fs,...
    length_sig,period_frac,sen_levels,sen_multiplier,min_level_zc)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% This program has been Edited to Assign the Generated Spikes so that the
% reconstruction work has less processing time.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% This program performs a simplified spike encoding of a simulated BM
% signal (usually this signal is created with a gammatone filterbank).
%
% The gammatone bank outputs multiple filter channels (parallel output,
```

```

% 'n_channels'), each of which is analysed by this program to produce a
% train of spikes.
% -->      bmSig(channel,data) where 'channel' is the channel number and
%          'data' is the data from that channel's filter
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% First we work out what the number of samples for the 1/4 cycle (in which
% to check the sensitivity levels and decide whether to produce a spike)
% will be for each filter channel

% First create an empty vector to store the sample values
cyc_samples = zeros(n_channels,1);

% Now loop over the number of filterbank channels
for i_cyc = 1:n_channels
    % There should be 'n_channels' number of sample values, one the 1/4
    % period for each filter channel
    %cyc_samples(i_cyc) = floor(period_frac*((1/f(i_cyc))*fs));
    cyc_samples(i_cyc) = floor(period_frac*((1/cochCFs(i_cyc))*fs));
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Create an empty vector, then fill in the lowest value threshold
threshold_level = zeros(sen_levels,1);
threshold_level(1) = min_level_zc;

% Loop over the number of sensitivity levels
for i_sen = 2:sen_levels
    % Increment each sensitivity level as the previous one X the

```

```

% multiplier
threshold_level(i_sen) = threshold_level(i_sen-1)*sen_multiplier;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This one generates spike in the convenient form for reconstruction.
% This one assigns Spike according to each channel. This one reduces
% processing time a lot.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Now we prepare the threshold levels to be used for the various
% sensitivities

% Now commence the big loop over each sensitivity level
h1 = waitbar(0,'Assigning Spikes according to Channels .....');
% Now we do the spike encoding, once for each sensitivity level
for ch = 1:n_channels

    % Create a counter to allow the index to run
    nn=0;
    ind_zc=[];

    % Now run the loop over the whole signal (up to the second to last
    % point to avoid trying to look at a data point that doesn't exist)
    for n = 1:length_sig-1

        % If the current sig level is less than zero and the next sample is
        % greater than zero... it's a zero crossing (count it at the nth
        % sample)
        if (bmSig(ch,n) < 0) && (bmSig(ch,n+1) >= 0)

            % Assign an initial 0 value to the to-be-extracted portion
            % of the signal
            sig_bit=0;

```



```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This is one is the original code written by Michel Newton, edited by me.
% This one assigns Spike according to each sensitivity level.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Now commence the big loop over each sensitivity level
h2 = waitbar(0,'Assigning Spikes according to Sensitivity Levels ...');

% Now we do the spike encoding, once for each sensitivity level
for i_thresh = 1:sen_levels

    % Create a cell array in which to store spikes for each channel
    zc_cell = cell(n_channels,1);
    for ch = 1:n_channels
        % Create a counter to allow the index to run
        nn=0;
        ind_zc=[];
        % Now run the loop over the whole signal (up to the second to last
        % point to avoid trying to look at a data point that doesn't exist)
        for n = 1:length_sig-1
            % If the current sig level is less than zero and the next sample is
            % greater than zero... it's a zero crossing (count it at the nth
            % sample)
            if (bmSig(ch,n) < 0) && (bmSig(ch,n+1) >= 0)
                % Assign an initial 0 value to the to-be-extracted portion
                % of the signal
                sig_bit=0;
                % Check the zero-crossing isn't too close to the start of
                % the signal (i.e. if there are enough samples over the
                % previous quarter cycle...)
                if (n+1) > cyc_samples(ch)
                    % Pick out the previous 1/4 cycle (absolute value)...
                    sig_bit = bmSig(ch,n+1-cyc_samples(ch):n+1);

```

```

        end

        % ... and compute the RMS value
        RMS = sqrt(sum(sig_bit.^2)/cyc_samples(ch));

        % Now we search through each sensitivity level, starting
        % with the most sensitive
        if RMS >= threshold_level(i_thresh)
            % If RMS value greater than any threshold level,
            % increment the counter 'nn' by 1...
            nn=nn+1;
            ind_zc(nn,1) = ch;
            ind_zc(nn,2) = (n+1)/fs;
            ind_zc(nn,3) = i_thresh;
        end
    end
end

zc_cell{ch} = ind_zc;
zc_all1 = cat(1,zc_cell{:});

% Here we sort the rows according to spike time, but only if spikes
% exist
switch isempty(zc_all1)
    case 0
        zc_all = sortrows(zc_all1,2);
    case 1
        zc_all = zc_all1;
end

% Here we put the data from each sensitivity level into a structure
% field
zc_struct(i_thresh).list = zc_all;

% Clear variables that run on each iteration of the sensitivity
% level
clear n nn sig_bit lev ind_zc zc_all1 zc_all

end

waitbar(i_thresh/sen_levels)

end

```

```
close(h2)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
end
```

Appendix C

Subjective Testing HTML Codes & Forms

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>Madhurananda Pahar's Home Page</title>
<meta name="keywords" content="" />
<meta name="description" content="" />
<meta name="author" content="Graham Cochrane" />
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<link rel="stylesheet" type="text/css" media="screen" href="/css/csams.css" />
<link rel="stylesheet" type="text/css" media="print" href="/css/csamp.css" />
</head>

<body>
    <h1>Sound Testing</h1>
    <h2>Please 'play' both sounds in a section and then answer the
question for each section. </h2>
    <form action ="questions_2.html">
    <br>
```



```
<table border="0" align="center">

<tr>

<td height="35"><b>Question 1: Which Sound is better?</b></td>

</tr>


<tr><td>

<h3>Sound 1a</h3>

<audio controls="controls" style="margin-top:10px;">

  <source src="http://www.cs.stir.ac.uk/~mpa/testing/sounds/CELESTA_F4.wav"

type="audio/wav" />

  Your browser does not support the audio element. Please

<a href="http://www.cs.stir.ac.uk/~mpa/testing/sounds/CELESTA_F4.wav">click here

</a> to play the sound.

</audio>

</td>

<td>

<h3>Sound 1b</h3>

<audio controls="controls" style="margin-top:10px;">

  <source src="http://www.cs.stir.ac.uk/~mpa/testing/sounds/

CELESTA_F4_16_50_NEW_Ramp_NoSpikeCap.wav" type="audio/wav" />

  Your browser does not support the audio element. Please <a href=

"http://www.cs.stir.ac.uk/~mpa/testing/sounds/

CELESTA_F4_16_50_NEW_Ramp_NoSpikeCap.wav">click here</a> to play the sound.

</audio>

</td></tr>


<tr>

<td height="65"><input type="submit" VALUE="Go to Question 2"/></td>

</tr>


</table>


</form>
```

```
<div id="footer">

<!-- Footer stuff goes below here -->


<!-- "#include" the validator images -->
<!--#include virtual="/include/validimages" -->
<!-- end of included validator images -->


<form action="/cgi-bin/get-email" method="post">
<address>
    Madhurananda Pahar ( Email: <input name="user" class="deptbutton"
        value="Madhurananda Pahar" type="submit" /> )<br />
    Room <a href="/intro/floorplan/?4B99">4B99</a>, Cottrell Building<br />
    Computing Science and Mathematics<br />
    School of Natural Sciences <br />
    University of Stirling
    Stirling FK9 4LA &nbsp; SCOTLAND<br />
</address>
</form>
</div>

<div id="footerbottom"></div>
</div>

<p id="footnote">
<a href="http://www.home.stir.ac.uk/cgi-bin/parser.pl">Text Only Site</a>
</p>
</body>
</html>
```

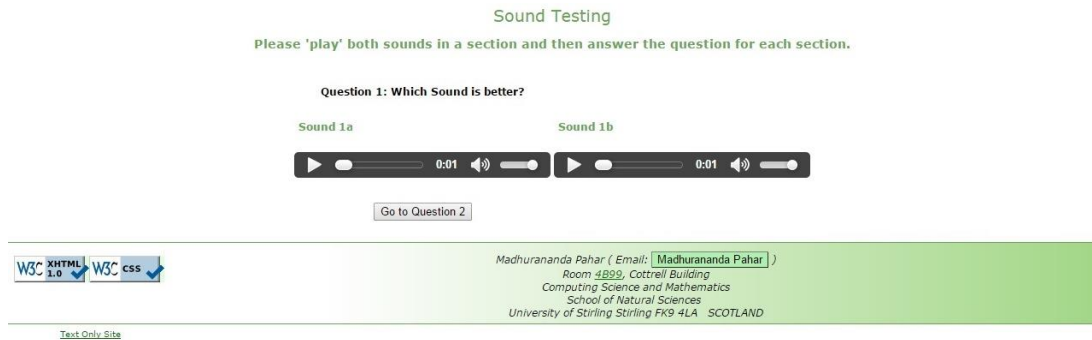


FIGURE C.1: **The Web Output of Question 1:-** This is the web output of the code mentioned above. The user has to choose the sound which is better among sound 1a and sound 1b.

Hello Everyone,

You are warmly invited to a Sound Testing Experiment as a part of an Audio Research Project in University of Stirling. This event is part of my PhD Thesis on Sound Coding and will not take more than 15 minutes. You can enjoy some sweets and crisps as some tribute to your voluntary contribution.

This experiment will take place in CS lab in 4B65 from 12 pm till 4 pm from Wednesday 23rd onwards. If you are interested, please come and see Madhu Pahar at Room 4B99 or send him an email at mpa@cs.stir.ac.uk.

Your willing voluntary spirit will be very much appreciated. The test has been posted online, to take a look please scan the code below –



Thank You.

FIGURE C.2: The Advert used to gather volunteers for subjective test.

User Information form

Please read and sign this form.

Name: _____

Age: _____ Gender: _____

Any known hearing problem? ☐ Yes ☐ No

If 'Yes', please provide details: _____

All the information provided here are accurate at the best of my knowledge.

Participant's Signature

Date:

Usability test consent form

Please read and sign this form.

You are invited to participate in a hearing test. In this hearing test session you will be asked to:

- Listen to a number of sounds wearing headphones.
- Answer the questions regarding each section.
- Tick your answer in a provided sheet.

Participation in this listening test session is voluntary. All information will remain strictly confidential. No identifiable information will be used. The data gathered will be used in a PhD thesis and possibly in a number of additional publications. However, at no time will your name or any other identification be used.

If you have any questions after today, please contact Madhu Pahar at mpa@cs.stir.ac.uk.

I have read and understood the information on this form and had all of my queries answered.

Participant's Signature

Date

Answer Script for Sound Test

Please answer all of the following questions at <http://www.cs.stir.ac.uk/~mpa/testing/index.html>. Attempt each Section after you hear that section containing two sounds.

Note: In this test, you are asked to choose a 'better' sound than other. Here the word 'better' means your understanding of the originality of that sound. Among any two sounds, please try to realise which sound is more natural and real. Select that sound as 'better' than other one.

If you struggle, please do not hesitate to ask.

Best Wishes

Question 1: Which Sound is better?

☐ Sound 1a ☐ Sound 1b

Question 2: Which Sound is better?

☐ Sound 2a ☐ Sound 2b

Question 3: Which Sound is better?

☐ Sound 3a ☐ Sound 3b

Question 4: Which Sound is better?

☐ Sound 4a ☐ Sound 4b

Question 5: Which Sound is better?

☐ Sound 5a ☐ Sound 5b

Question 6: Which Sound is better?

☐ Sound 6a ☐ Sound 6b

Question 7: Which Sound is better?

☐ Sound 7a ☐ Sound 7b

Question 8: Which Sound is better?

☐ Sound 8a ☐ Sound 8b

Question 9: Which Sound is better?

☐ Sound 9a ☐ Sound 9b

Question 10: Which Sound is better?

☐ Sound 10a ☐ Sound 10b

Question 11: Which Sound is better?

☐ Sound 11a ☐ Sound 11b

Question 12: Which Sound is better?

☐ Sound 12a ☐ Sound 12b

Question 13: Which Sound is better?

☐ Sound 13a ☐ Sound 13b

Question 14: Which Sound is better?

☐ Sound 14a ☐ Sound 14b

Question 15: Which Sound is better?

☐ Sound 15a ☐ Sound 15b

Question 16: Which Sound is better?

☐ Sound 16a ☐ Sound 16b

Question 17: Which Sound is better?

☐ Sound 17a ☐ Sound 17b

Question 18: Which Sound is better?

☐ Sound 18a ☐ Sound 18b

Question 19: Which Sound is better?

☐ Sound 19a ☐ Sound 19b

Question 20: Which Sound is better?

☐ Sound 20a ☐ Sound 20b

Participant's Signature

Date

Appendix D

Perceptual Evaluation of Speech Quality & Composite Test (MATLAB Codes)

```
function [Csig,Cbak,Covl]= composite(cleanFile, enhancedFile);

% ----- composite objective measure -----
%
%   Center for Robust Speech Systems
%   University of Texas-Dallas
%   Copyright (c) 2006
%   All Rights Reserved.
%
% Description:
%
%   This function implements the composite objective measure
%   proposed in [1]. It returns three values: The predicted rating of
%   overall quality (Covl), the rating of speech distortion (Csig) and
%   the rating of background distortion (Cbak). The ratings are based on
%   the 1-5 MOS scale.
```

```

%      In addition, it returns the values of the SNRseg, log-likelihood
%      ratio (LLR), PESQ
%      and weighted spectral slope (WSS) objective measures.
%
%      References:
%      [1] Hu, Y. and Loizou, P. (2006). \Evaluation of objective measures
%      for speech enhancement,
%          Proceedings of INTERSPEECH-2006, Philadelphia, PA, September 2006.
%
%
%      Authors:
%          Philipos C. Loizou and Yi Hu
%          Bryan L. Pellom and John H. L. Hansen (for the implementation of
%          the WSS, LLR and SnrSeg measures)
%
%-----

if nargin<2
    fprintf('Usage: [Csig,Cbak,Covl]=composite(cleanfile.wav,enhanced.wav)\n');
    fprintf('where ''Csig'' is the predicted rating of speech distortion\n');
    fprintf('      ''Cbak'' is the predicted rating of background distortion\n');
    fprintf('      ''Covl'' is the predicted rating of overall quality.\n\n');
    return;
end

alpha= 0.95;

[data1, Srate1, Nbits1]= wavread(cleanFile);
[data2, Srate2, Nbits2]= wavread(enhancedFile);
if ( Srate1~= Srate2) | ( Nbits1~= Nbits2)
    error( 'The two files do not match!\n');
end

```

```

len= min( length( data1), length( data2));
data1= data1( 1: len)+eps;
data2= data2( 1: len)+eps;

% -- compute the WSS measure ---
%
wss_dist_vec= wss( data1, data2,Srate1);
wss_dist_vec= sort( wss_dist_vec);
wss_dist= mean( wss_dist_vec( 1: round( length( wss_dist_vec)*alpha)));

% --- compute the LLR measure -----
%
LLR_dist= llr( data1, data2,Srate1);
LLRs= sort(LLR_dist);
LLR_len= round( length(LLR_dist)* alpha);
llr_mean= mean( LLRs( 1: LLR_len));

% --- compute the SNRseg -----
%
[snr_dist, segsnr_dist]= snr( data1, data2,Srate1);
snr_mean= snr_dist;
segSNR= mean( segsnr_dist);

% -- compute the pesq ----
[pesq_mos]= pesq(Srate1,cleanFile, enhancedFile);

% --- now compute the composite measures -----
%
Csig = 3.093 - 1.029*llr_mean + 0.603*pesq_mos-0.009*wss_dist;
Cbak = 1.634 + 0.478 *pesq_mos - 0.007*wss_dist + 0.063*segSNR;
Covl = 1.594 + 0.805*pesq_mos - 0.512*llr_mean - 0.007*wss_dist;

```

```
fprintf('\n LLR=%f    SNRseg=%f    WSS=%f    PESQ=%f\n',llr_mean,segSNR,...
        wss_dist,pesq_mos);

return;

% -----
%
%   Weighted Spectral Slope (WSS) Objective Speech Quality Measure
%
%   Center for Robust Speech Systems
%   University of Texas-Dallas
%   Copyright (c) 1998-2006
%   All Rights Reserved.
%
% Description:
%
%   This function implements the Weighted Spectral Slope (WSS)
%   distance measure originally proposed in [1]. The algorithm
%   works by first decomposing the speech signal into a set of
%   frequency bands (this is done for both the test and reference
%   frame). The intensities within each critical band are
%   measured. Then, a weighted distances between the measured
%   slopes of the log-critical band spectra are computed.
%   This measure is also described in Section 2.2.9 (pages 56-58)
%   of [2].
%
%   Whereas Klatt's original measure used 36 critical-band
%   filters to estimate the smoothed short-time spectrum, this
%   implementation considers a bank of 25 filters spanning
%   the 4 kHz bandwidth.
%
% Input/Output:
%
```

```
% The input is a reference 8kHz sampled speech, and processed
% speech (could be noisy or enhanced).
%
% The function returns the numerical distance between each
% frame of the two input files (one distance per frame).
%
% References:
%
% [1] D. H. Klatt, "Prediction of Perceived Phonetic Distance
% from Critical-Band Spectra: A First Step", Proc. IEEE
% ICASSP'82, Volume 2, pp. 1278-1281, May, 1982.
%
% [2] S. R. Quackenbush, T. P. Barnwell, and M. A. Clements,
% Objective Measures of Speech Quality. Prentice Hall
% Advanced Reference Series, Englewood Cliffs, NJ, 1988,
% ISBN: 0-13-629056-6.
%
% Authors:
%
% Bryan L. Pellom and John H. L. Hansen
%
%
% Last Modified:
%
% July 22, 1998
% September 12, 2006 by Philipos Loizou
% -----

function distortion = wss(clean_speech, processed_speech,sample_rate)

% -----
% Check the length of the clean and processed speech. Must be the same.
% -----
```

```

clean_length      = length(clean_speech);
processed_length  = length(processed_speech);

if (clean_length ~= processed_length)
    disp('Error: Files must have same length.');
```

```

    return
end

% -----
% Global Variables
% -----

% sample_rate = 8000;    % default sample rate
% winlength   = 240;    % window length in samples
% skiprate    = 60;     % window skip in samples
winlength     = round(30*sample_rate/1000); %240;    % window length in samples
skiprate      = floor(winlength/4);    % window skip in samples
max_freq      = sample_rate/2;    % maximum bandwidth
num_crit      = 25;    % number of critical bands

USE_FFT_SPECTRUM = 1;    % defaults to 10th order LP spectrum
% n_fft       = 512;    % FFT size
n_fft         = 2^nextpow2(2*winlength);
n_fftby2      = n_fft/2;    % FFT size/2
Kmax          = 20;    % value suggested by Klatt, pg 1280
Klocmax       = 1;    % value suggested by Klatt, pg 1280

% -----
% Critical Band Filter Definitions (Center Frequency and Bandwidths in Hz)
% -----

```

```

cent_freq(1) = 50.0000;   bandwidth(1) = 70.0000;
cent_freq(2) = 120.000;   bandwidth(2) = 70.0000;
cent_freq(3) = 190.000;   bandwidth(3) = 70.0000;
cent_freq(4) = 260.000;   bandwidth(4) = 70.0000;
cent_freq(5) = 330.000;   bandwidth(5) = 70.0000;
cent_freq(6) = 400.000;   bandwidth(6) = 70.0000;
cent_freq(7) = 470.000;   bandwidth(7) = 70.0000;
cent_freq(8) = 540.000;   bandwidth(8) = 77.3724;
cent_freq(9) = 617.372;   bandwidth(9) = 86.0056;
cent_freq(10) = 703.378;   bandwidth(10) = 95.3398;
cent_freq(11) = 798.717;   bandwidth(11) = 105.411;
cent_freq(12) = 904.128;   bandwidth(12) = 116.256;
cent_freq(13) = 1020.38;   bandwidth(13) = 127.914;
cent_freq(14) = 1148.30;   bandwidth(14) = 140.423;
cent_freq(15) = 1288.72;   bandwidth(15) = 153.823;
cent_freq(16) = 1442.54;   bandwidth(16) = 168.154;
cent_freq(17) = 1610.70;   bandwidth(17) = 183.457;
cent_freq(18) = 1794.16;   bandwidth(18) = 199.776;
cent_freq(19) = 1993.93;   bandwidth(19) = 217.153;
cent_freq(20) = 2211.08;   bandwidth(20) = 235.631;
cent_freq(21) = 2446.71;   bandwidth(21) = 255.255;
cent_freq(22) = 2701.97;   bandwidth(22) = 276.072;
cent_freq(23) = 2978.04;   bandwidth(23) = 298.126;
cent_freq(24) = 3276.17;   bandwidth(24) = 321.465;
cent_freq(25) = 3597.63;   bandwidth(25) = 346.136;

```

```

bw_min      = bandwidth (1);    % minimum critical bandwidth

```

```

% -----
% Set up the critical band filters. Note here that Gaussianly shaped
% filters are used. Also, the sum of the filter weights are equivalent
% for each critical band filter. Filter less than -30 dB and set to
% zero.
% -----

```

```

min_factor = exp (-30.0 / (2.0 * 2.303));          % -30 dB point of filter

for i = 1:num_crit
    f0 = (cent_freq (i) / max_freq) * (n_fftby2);
    all_f0(i) = floor(f0);
    bw = (bandwidth (i) / max_freq) * (n_fftby2);
    norm_factor = log(bw_min) - log(bandwidth(i));
    j = 0:1:n_fftby2-1;
    crit_filter(i,:) = exp (-11 *(((j - floor(f0)) ./bw).^2) + norm_factor);
    crit_filter(i,:) = crit_filter(i,:).*(crit_filter(i,:) > min_factor);
end

% -----
% For each frame of input speech, calculate the Weighted Spectral
% Slope Measure
% -----

num_frames = clean_length/skiprate-(winlength/skiprate); % number of frames
start      = 1; % starting sample
window     = 0.5*(1 - cos(2*pi*(1:winlength)'/(winlength+1)));

for frame_count = 1:num_frames

    % -----
    % (1) Get the Frames for the test and reference speech.
    %     Multiply by Hanning Window.
    % -----

    clean_frame = clean_speech(start:start+winlength-1);
    processed_frame = processed_speech(start:start+winlength-1);
    clean_frame = clean_frame.*window;
    processed_frame = processed_frame.*window;

```



```

% -----
% (2) Compute the Power Spectrum of Clean and Processed
% -----

if (USE_FFT_SPECTRUM)
    clean_spec      = (abs(fft(clean_frame,n_fft)).^2);
    processed_spec = (abs(fft(processed_frame,n_fft)).^2);
else
    a_vec = zeros(1,n_fft);
    a_vec(1:11) = lpc(clean_frame,10);
    clean_spec      = 1.0/(abs(fft(a_vec,n_fft)).^2)';

    a_vec = zeros(1,n_fft);
    a_vec(1:11) = lpc(processed_frame,10);
    processed_spec = 1.0/(abs(fft(a_vec,n_fft)).^2)';
end

% -----
% (3) Compute Filterbank Output Energies (in dB scale)
% -----

for i = 1:num_crit
    clean_energy(i) = sum(clean_spec(1:n_fftby2) ...
        .*crit_filter(i,:))';
    processed_energy(i) = sum(processed_spec(1:n_fftby2) ...
        .*crit_filter(i,:))';
end

clean_energy = 10*log10(max(clean_energy,1E-10));
processed_energy = 10*log10(max(processed_energy,1E-10));

% -----
% (4) Compute Spectral Slope (dB[i+1]-dB[i])
% -----

```

```

clean_slope      = clean_energy(2:num_crit) - ...
    clean_energy(1:num_crit-1);
processed_slope = processed_energy(2:num_crit) - ...
    processed_energy(1:num_crit-1);

% -----
% (5) Find the nearest peak locations in the spectra to
%     each critical band.  If the slope is negative, we
%     search to the left.  If positive, we search to the
%     right.
% -----

for i = 1:num_crit-1

    % find the peaks in the clean speech signal

    if (clean_slope(i)>0) % search to the right
n = i;
        while ((n<num_crit) & (clean_slope(n) > 0))
            n = n+1;
        end
        clean_loc_peak(i) = clean_energy(n-1);
    else % search to the left
        n = i;
        while ((n>0) & (clean_slope(n) <= 0))
            n = n-1;
        end
        clean_loc_peak(i) = clean_energy(n+1);
    end

    % find the peaks in the processed speech signal

    if (processed_slope(i)>0) % search to the right
n = i;

```

```

        while ((n<num_crit) & (processed_slope(n) > 0))
            n = n+1;
        end
        processed_loc_peak(i) = processed_energy(n-1);
        else % search to the left
            n = i;
        while ((n>0) & (processed_slope(n) <= 0))
            n = n-1;
        end
        processed_loc_peak(i) = processed_energy(n+1);
        end

    end

% -----
% (6) Compute the WSS Measure for this frame. This
%      includes determination of the weighting function.
% -----

dBMax_clean      = max(clean_energy);
dBMax_processed  = max(processed_energy);

% The weights are calculated by averaging individual
% weighting factors from the clean and processed frame.
% These weights W_clean and W_processed should range
% from 0 to 1 and place more emphasis on spectral
% peaks and less emphasis on slope differences in spectral
% valleys. This procedure is described on page 1280 of
% Klatt's 1982 ICASSP paper.

Wmax_clean      = Kmax ./ (Kmax + dBMax_clean - ...
    clean_energy(1:num_crit-1));
Wlocmax_clean    = Klocmax ./ ( Klocmax + clean_loc_peak - ...
    clean_energy(1:num_crit-1));

```

```

W_clean          = Wmax_clean .* Wlocmax_clean;

Wmax_processed    = Kmax ./ (Kmax + dBMax_processed - ...
    processed_energy(1:num_crit-1));
Wlocmax_processed = Klocmax ./ ( Klocmax + processed_loc_peak - ...
    processed_energy(1:num_crit-1));
W_processed       = Wmax_processed .* Wlocmax_processed;

W = (W_clean + W_processed)./2.0;

distortion(frame_count) = sum(W.*(clean_slope(1:num_crit-1) - ...
    processed_slope(1:num_crit-1)).^2);

% this normalization is not part of Klatt's paper, but helps
% to normalize the measure. Here we scale the measure by the
% sum of the weights.

distortion(frame_count) = distortion(frame_count)/sum(W);

start = start + skiprate;

end

%-----
function distortion = llr(clean_speech, processed_speech,sample_rate)

% -----
% Check the length of the clean and processed speech. Must be the same.
% -----

clean_length      = length(clean_speech);
processed_length   = length(processed_speech);

```

```

if (clean_length ~= processed_length)
    disp('Error: Both Speech Files must be same length.');
```

```

    return
end

% -----
% Global Variables
% -----

% sample_rate = 8000;    % default sample rate
% winlength    = 240;    % window length in samples
% skiprate     = 60;    % window skip in samples
% P            = 10;    % LPC Analysis Order
winlength      = round(30*sample_rate/1000); % window length in samples
skiprate       = floor(winlength/4);    % window skip in samples
if sample_rate<10000
    P           = 10;    % LPC Analysis Order
else
    P=16;        % this could vary depending on sampling frequency.
end

% -----
% For each frame of input speech, calculate the Log Likelihood Ratio
% -----

num_frames = clean_length/skiprate-(winlength/skiprate); % number of frames
start      = 1; % starting sample
window     = 0.5*(1 - cos(2*pi*(1:winlength)'/(winlength+1)));

for frame_count = 1:num_frames

    % -----
    % (1) Get the Frames for the test and reference speech.
    %     Multiply by Hanning Window.
```

```

% -----

clean_frame = clean_speech(start:start+winlength-1);
processed_frame = processed_speech(start:start+winlength-1);
clean_frame = clean_frame.*window;
processed_frame = processed_frame.*window;

% -----
% (2) Get the autocorrelation lags and LPC parameters used
%      to compute the LLR measure.
% -----

[R_clean, Ref_clean, A_clean] = ...
    lpcoeff(clean_frame, P);
[R_processed, Ref_processed, A_processed] = ...
    lpcoeff(processed_frame, P);

% -----
% (3) Compute the LLR measure
% -----

numerator    = A_processed*toeplitz(R_clean)*A_processed';
denominator  = A_clean*toeplitz(R_clean)*A_clean';
distortion(frame_count) = log(numerator/denominator);
start = start + skiprate;

end

%-----
function [acorr, refcoeff, lpparams] = lpcoeff(speech_frame, model_order)

% -----
% (1) Compute Autocorrelation Lags
% -----

```

```

winlength = max(size(speech_frame));
for k=1:model_order+1
    R(k) = sum(speech_frame(1:winlength-k+1) ...
        .*speech_frame(k:winlength));
end

% -----
% (2) Levinson-Durbin
% -----

a = ones(1,model_order);
E(1)=R(1);
for i=1:model_order
    a_past(1:i-1) = a(1:i-1);
    sum_term = sum(a_past(1:i-1).*R(i:-1:2));
    rcoeff(i)=(R(i+1) - sum_term) / E(i);
    a(i)=rcoeff(i);
    a(1:i-1) = a_past(1:i-1) - rcoeff(i).*a_past(i-1:-1:1);
    E(i+1)=(1-rcoeff(i)*rcoeff(i))*E(i);
end

acorr      = R;
refcoeff   = rcoeff;
lpparams   = [1 -a];

% -----

function [overall_snr, segmental_snr] = snr(clean_speech, processed_speech,sample_rate)

% -----
% Check the length of the clean and processed speech.  Must be the same.
% -----

```

```

clean_length      = length(clean_speech);
processed_length  = length(processed_speech);

if (clean_length ~= processed_length)
    disp('Error: Both Speech Files must be same length.');
```

```

    return
end

% -----
% Scale both clean speech and processed speech to have same dynamic
% range. Also remove DC component from each signal
% -----

%clean_speech      = clean_speech      - mean(clean_speech);
%processed_speech  = processed_speech - mean(processed_speech);

%processed_speech = processed_speech.*(max(abs(clean_speech))/ max(abs(processed_spee

overall_snr = 10* log10( sum(clean_speech.^2)/sum((...
    clean_speech-processed_speech).^2));

% -----
% Global Variables
% -----

% sample_rate = 8000;    % default sample rate
% winlength   = 240;    % window length in samples
% skiprate    = 60;     % window skip in samples
winlength     = round(30*sample_rate/1000); %240;    % window length in samples
skiprate      = floor(winlength/4);    % window skip in samples
MIN_SNR       = -10;    % minimum SNR in dB
MAX_SNR       = 35;     % maximum SNR in dB

```



```

% -----
% For each frame of input speech, calculate the Segmental SNR
% -----

num_frames = clean_length/skiprate-(winlength/skiprate); % number of frames
start      = 1; % starting sample
window     = 0.5*(1 - cos(2*pi*(1:winlength)'/(winlength+1)));

for frame_count = 1: num_frames

    % -----
    % (1) Get the Frames for the test and reference speech.
    %     Multiply by Hanning Window.
    % -----

    clean_frame = clean_speech(start:start+winlength-1);
    processed_frame = processed_speech(start:start+winlength-1);
    clean_frame = clean_frame.*window;
    processed_frame = processed_frame.*window;

    % -----
    % (2) Compute the Segmental SNR
    % -----

    signal_energy = sum(clean_frame.^2);
    noise_energy  = sum((clean_frame-processed_frame).^2);
    segmental_snr(frame_count) = 10*log10(signal_energy/(noise_energy+eps)+eps);
    segmental_snr(frame_count) = max(segmental_snr(frame_count),MIN_SNR);
    segmental_snr(frame_count) = min(segmental_snr(frame_count),MAX_SNR);

    start = start + skiprate;

end

```

Bibliography

- [1] Oxford. Spike definition, January 2016. URL <http://www.oxforddictionaries.com/definition/english/spike>. Date Accessed: March 2016.
- [2] Oxford Dictionary. Event definition, January 2016. URL <http://www.oxforddictionaries.com/definition/english/event>. Date Accessed: March 2016.
- [3] Leslie S Smith and Dagmar S Fraser. Robust sound onset detection using leaky integrate-and-fire neurons with depressing synapses. *Neural Networks, IEEE Transactions on*, 15(5):1125–1134, 2004.
- [4] L.S. Smith and S. Collins. Determining ITDs using two microphones on a flat panel during onset intervals with a biologically inspired spike-based technique. *Audio, Speech, and Language Processing, IEEE Transactions on*, 15(8):2278–2286, nov. 2007. ISSN 1558-7916. doi: 10.1109/TASL.2007.904212.
- [5] Luiz Carlos Gouveia, Thomas Jacob Koickal, and Alister Hamilton. An asynchronous spike event coding scheme for programmable analog arrays. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 58(4):791–799, 2011.
- [6] Michael Purdy and Deborah Borisoff. *Listening in everyday life: A personal and professional approach*. University Press of America, 1996.
- [7] David Morton. The history of sound recording, January 2016. URL <http://www.recording-history.org/HTML/musictech1.php>. The Sound Recording Technology History Site explores the history and impact of the inventions that changed the way we listen. Date Accessed: March 2016.
- [8] David Morton. The history of sound recording, January 2016. URL <http://www.recording-history.org/HTML/musictech4.php>. Date Accessed: March 2016.

- [9] David Morton. The history of sound recording, January 2016. URL <http://www.recording-history.org/HTML/musictech7.php>. Date Accessed: March 2016.
- [10] David Morton. The history of sound recording, January 2016. URL <http://www.recording-history.org/HTML/musictech11.php>. Date Accessed: March 2016.
- [11] David English. Sound blaster turns pro. Compute Issue 141, June 1992. URL http://www.atarimagazines.com/compute/issue141/82_Sound_Blaster_turns_.php. Page 82. Date Accessed: March 2016.
- [12] Karlheinz Brandenburg. MP3 and AAC explained. In *Audio Engineering Society Conference: 17th International Conference: High-Quality Audio Coding*. Audio Engineering Society, 1999.
- [13] Mario A Ruggiero. Physiology and coding of sound in the auditory nerve. In *The mammalian auditory pathway: Neurophysiology*, pages 34–93. Springer, 1992.
- [14] Federica Bianchi, Sébastien Santurette, Michal Fereczkowski, and Torsten Dau. Relation between temporal envelope coding, pitch discrimination, and compression estimates in listeners with sensorineural hearing loss. *Acoustical Society of America Journal*, 137(4), 2015. ISSN 0001-4966. doi: 10.1121/1.4920124.
- [15] Morten Løve Jepsen and Torsten Dau. Modeling auditory perception of individual hearing-impaired listeners, 2008.
- [16] P Cosi. Auditory modelling for speech analysis and recognition. *Visual Representation of Speech Signals*, Wiley & Sons Chichester, pages pp. 205–212, 1993.
- [17] Piero Cosi, Enrico Zovato, Centro di Studi per le Ricerche, and CNR di Fonetica. Lyon’s auditory model inversion: a tool for sound separation and speech enhancement. In *Proceedings of ESCA Workshop on ‘The Auditory Basis of Speech Perception’*, Keele University, Keele (UK), pages 194 – 197. Citeseer, 1996.
- [18] Hiroshi G Okuno, Tomohiro Nakatani, and Takeshi Kawabata. Interfacing sound stream segregation to automatic speech recognition-preliminary results on listening to several sounds simultaneously. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1082–1089. Citeseer, 1996.

- [19] ReSound Alera. Resound Alera audiology background. URL <http://www.resound.com/~media/DownloadLibrary/ReSound/Products/Alera/resound-alera-audiology-background.pdf>. Date Accessed: March 2016.
- [20] Rassol Raissi. The theory behind mp3. *MP3'Tech*, 2002.
- [21] Marina Bosi, Karlheinz Brandenburg, Schuyler Quackenbush, Louis Fielder, Kenzo Akagiri, Hendrik Fuchs, and Martin Dietz. ISO/IEC MPEG-2 advanced audio coding. *Journal of the Audio engineering society*, 45(10):789–814, 1997.
- [22] Michael A Gerzon, Peter G Craven, J Robert Stuart, Malcolm J Law, and Rhonda J Wilson. The mlp lossless compression system. In *Audio Engineering Society Conference: 17th International Conference: High-Quality Audio Coding*. Audio Engineering Society, 1999.
- [23] Brian CJ Moore. *An introduction to the psychology of hearing*. Brill, 2012.
- [24] Auditory system: 12 cranial nerves, January 2016. URL <https://12cranialnerves.files.wordpress.com/2012/04/vestibulocochlear-nerve.jpg>. Date Accessed: March, 2016.
- [25] Bruce Masterton, Henry Heffner, and Richard Ravizza. The evolution of human hearing. *The Journal of the Acoustical Society of America*, 45(4):966–985, 1969.
- [26] The McGraw-Hill Companies. Auditory nerves, January 2016. URL https://classconnection.s3.amazonaws.com/766/flashcards/182766/png/auditory_&_vestibular_system1320616096588.png. Date Accessed: March 2016.
- [27] Professor of Psychology David Heeger and Neural Science. Traveling wave, January 2016. URL <http://www.cns.nyu.edu/~david/courses/perception/lecturenotes/pitch/pitch-slides/Slide2.jpg>. Date Accessed: March 2016.
- [28] Georg Von Békésy and Ernest Glen Wever. *Experiments in hearing*, volume 8. McGraw-Hill New York, 1960.
- [29] James O Pickles. *An introduction to the physiology of hearing*. Brill, 2012.
- [30] BM Johnstone, R Patuzzi, and GK Yates. Basilar membrane measurements and the travelling wave. *Hearing research*, 22(1):147–153, 1986.

- [31] Mario A Ruggero, Nola C Rich, Alberto Recio, S Shyamla Narayan, and Luis Robles. Basilar-membrane responses to tones at the base of the chinchilla cochlea. *The Journal of the Acoustical Society of America*, 101(4):2151–2163, 1997.
- [32] Alberto Recio-Spinoso and Enrique A Lopez-Poveda. Basilar membrane responses to simultaneous presentations of white noise and a single tone. In *The Neurophysiological Bases of Auditory Perception*, pages 15–23. Springer, 2010.
- [33] Ph.D. Janet Lyn Fitzakerley. Displacement of bm, January 2016. URL <http://www.d.umn.edu/~jfitzake/Lectures/DMED/InnerEar/CochlearPhysiology/Figures/PlacePrinciple.png>. Date Accessed: March 2016.
- [34] Henry Vandyke Carter. Anatomy of the human body, January 1958. URL <https://upload.wikimedia.org/wikipedia/commons/9/9f/Gray931.png>. Bartleby.com: Gray’s Anatomy, Plate 931. Date Accessed: March 2016.
- [35] Beyond the Dish. Cross-section of cochlea, February 2012. URL <https://beyondthedish.files.wordpress.com/2012/02/cochlea-cross-section.jpg>. Date Accessed: March 2016.
- [36] Redted71. Cochlear nucleus innervated by a branching auditory nerve fibre, 4 December 2006. URL https://upload.wikimedia.org/wikipedia/en/0/0b/Cochlear_nucleus_innervated_by_a_branching_auditory_nerve_fibre.JPG. Drawn using Microsoft Paint. Date Accessed: March 2016.
- [37] Roy D Patterson, K Robinson, J Holdsworth, D McKeown, C Zhang, and M Allershand. Complex sounds and auditory images. *Auditory physiology and perception*, 83:429–446, 1992.
- [38] Martin Cooke. *Modelling Auditory Processing and Organisation*. Cambridge University Press, 1933.
- [39] Egbert De Boer and Paul Kuyper. Triggered correlation. *Biomedical Engineering, IEEE Transactions on*, (3):169–179, 1968.
- [40] E De Boer and HR De Jongh. On cochlear encoding: Potentialities and limitations of the reverse-correlation technique. *The Journal of the Acoustical Society of America*, 63(1):115–135, 1978.

- [41] Steeb W-H Stoop N Stoop R Kern A, Heid C. Biophysical parameters modification could overcome essential hearing gaps, 29 August 2008. URL https://upload.wikimedia.org/wikipedia/commons/6/65/Uncoiled_cochlea_with_basilar_membrane.png. The position x of the maximal amplitude of the travelling wave corresponds in a 1-to-1 way to a stimulus frequency. Date Accessed: March 2016.
- [42] The State University of New Jersey Dr. Uzwiak. Audition files: Basilar membrane, January 2016. URL http://www.rci.rutgers.edu/~uzwiak/AnatPhys/Audition_files/image012.jpg. Date Accessed: March 2016.
- [43] Rick Fofi. Hearing: Basilar membrane, January 2016. URL <http://www.austincc.edu/rfofi/NursingRvw/NursingPics/PNSafferentpt2Pics/Picture23.jpg>. Date Accessed: March 2016.
- [44] RD Patterson, Ian Nimmo-Smith, John Holdsworth, and Peter Rice. An efficient auditory filterbank based on the gammatone function. In *A meeting of the IOC Speech Group on Auditory Modelling at RSRE*, volume 2, 1987.
- [45] John Holdsworth, Ian Nimmo-Smith, Roy Patterson, and Peter Rice. Implementing a gammatone filter bank. *Annex C of the SVOS Final Report: Part A: The Auditory Filterbank*, 1:1–5, 1988.
- [46] Andreas G Katsiamis, Emmanuel M Drakakis, and Richard F Lyon. Practical gammatone-like filters for auditory processing. *EURASIP Journal on Audio, Speech, and Music Processing*, 2007(1):063685, 2007.
- [47] Malcolm Slaney, Daniel Naar, and Richard F Lyon. Auditory model inversion for sound separation. In *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, volume 2, pages II–77. IEEE, 1994.
- [48] Rémi Decorsière, Peter L Søndergaard, Ewen N MacDonald, and Torsten Dau. Inversion of auditory spectrograms, traditional spectrograms, and other envelope representations. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, 23(1):46–56, 2015.
- [49] Christian J Sumner, Enrique A Lopez-Poveda, Lowel P O’Mard, and Ray Meddis. Adaptation in a revised inner-hair cell model. *The Journal of the Acoustical Society of America*, 113(2):893 – 901, 2003.

- [50] Eliathamby Ambikairajah, Julien Epps, and Lee Lin. Wideband speech and audio coding using gammatone filter banks. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, volume 2, pages 773–776. IEEE, 2001.
- [51] Leslie S Smith. Toward a neuromorphic microphone. *Frontiers in neuroscience*, 9, 2015.
- [52] Oxford. Onset definition, January 2016. URL <http://www.oxforddictionaries.com/definition/english/onset>. Date Accessed: March 2016.
- [53] Shih-Chii Liu, Tobi Delbruck, Giacomo Indiveri, Adrian Whatley, and Rodney Douglas. Silicon cochleas. *Event-Based Neuromorphic Systems*, pages 71–90, 2014.
- [54] Shiwei Wang, Thomas Jacob Koickal, Godwin Enemali, Luiz Gouveia, Lei Wang, and Alister Hamilton. Design of a silicon cochlea system with biologically faithful response. In *Neural Networks (IJCNN), 2015 International Joint Conference on*, pages 1–7. IEEE, 2015.
- [55] International Electrotechnical Commission et al. *Information Technology: Coding of Moving Pictures and Associated Audio for Digital Storage Media at Up to about 1, 5 Mbit/s*. ISO/IEC, 1993.
- [56] Ryan Bemrose. Louder sounds better, 7 February 2007. URL http://blogs.msdn.com/blogfiles/audiofool/WindowsLiveWriter/LouderSoundsBetter_12855/FletcherMunson_EqualLoudness2.jpg. Date Accessed: March 2016.
- [57] Qihai Zhou. *Theoretical and Mathematical Foundations of Computer Science: Second International Conference, ICTMF 2011, Singapore, May 5-6, 2011, Revised Selected Papers*, volume 164. Springer Science & Business Media, 2011.
- [58] Rob Koenen. MPEG-4 multimedia for our time. *Spectrum, IEEE*, 36(2):26–33, 1999.
- [59] Tilman Liebchen, Takehiro Moriya, Noboru Harada, Yutaka Kamamoto, and Yuriy A Reznik. The MPEG-4 Audio Lossless Coding (ALS) standard-technology and applications. In *AES 119th Convention paper*, 2005.

- [60] Guo-Hong Gao, Xue-Yong Li, Shi-Tao Yan, and Jin-Na Lv. The design and implementation of MP4 coding system based on s3c2410. In *Theoretical and Mathematical Foundations of Computer Science*, pages 579–583. Springer, 2011.
- [61] Stefan Meltzer and Gerald Moser. Mpeg-4 he-aac v2-audio coding for today’s digital media world. *EBU technical Review*, 305:37–38, 2006.
- [62] Tony Smith. Microsoft readies MP3-killer digital music format, March 1999. URL http://www.theregister.co.uk/1999/03/12/microsoft_readies_mp3killer_digital_music/. Date Accessed: March 2016.
- [63] Windows media 9 series capabilities and benefits overview, Retrieved 2007-08-16. URL <http://www.cse.dmu.ac.uk/~hoi/mult2003/week6/WinMedia9WhitePaper.doc>. https://en.wikipedia.org/wiki/Windows_Media_Audio#cite_note-Windows_Media_9_Series_Whitepaper-3. Date Accessed: March 2016.
- [64] Andrew Abel and Amir Hussain. Novel two-stage audiovisual speech filtering in noisy environments. *Cognitive Computation*, 6(2):200–217, 2014.
- [65] Yi Hu and Philipos C Loizou. Evaluation of objective measures for speech enhancement. In *Interspeech*, 2006.
- [66] Yi Hu and Philipos C Loizou. Evaluation of objective quality measures for speech enhancement. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(1):229–238, 2008.
- [67] Dennis Klatt. Prediction of perceived phonetic distance from critical-band spectra: A first step. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP’82.*, volume 7, pages 1278–1281. IEEE, 1982.
- [68] Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra. Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP’01). 2001 IEEE International Conference on*, volume 2, pages 749–752. IEEE, 2001.
- [69] Schuyler R Quackenbush, Thomas Pinkney Barnwell, and Mark A Clements. *Objective measures of speech quality*. Prentice Hall, 1988.

- [70] ITU P.862. An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs, October 2013. URL <http://www.itu.int/rec/T-REC-P.862/en>. Date Accessed: March 2016.
- [71] Sheila Flanagan, Brian CJ Moore, and Michael A Stone. Discrimination of group delay in clicklike signals presented via headphones and loudspeakers. *Journal of the Audio Engineering Society*, 53(7/8):593–611, 2005.
- [72] John S Garofolo, Linguistic Data Consortium, et al. *TIMIT: acoustic-phonetic continuous speech corpus*. Linguistic Data Consortium, 1993. URL <https://catalog.ldc.upenn.edu/LDC93S1>. Date Accessed: March 2016.
- [73] Ulf Grenander. *Probability and Statistics: The Harald Cram Volume*. The Nyquist frequency is that frequency whose period is two sampling intervals. Alqvist & Wiksell, 1959.
- [74] Harry L Stiltz. *Aerospace telemetry*, volume 1 of *the existence of power in the continuous signal spectrum at frequencies higher than the Nyquist frequency is the cause of aliasing error*. Prentice-Hall, 1961.
- [75] Donata Oertel, Ramazan Bal, Stephanie M Gardner, Philip H Smith, and Philip X Joris. Detection of synchrony in the activity of auditory nerve fibers by octopus cells of the mammalian cochlear nucleus. *Proceedings of the National Academy of Sciences*, 97(22):11773–11779, 2000.
- [76] Fabio S. Flosi. Father landell de moura - radio broadcasting pioneer, 2012. URL <http://www.aminharadio.com/radio/files/Artigo-Revista-PCP-USA.pdf>. fabioflosi@hotmail.com. Date Accessed: March 2016.
- [77] John Bray. *Innovation and the communications revolution: from the Victorian pioneers to broadband Internet*. Iet, 2002.
- [78] Jordan Santell. Signal processing with web audio, March 2016. URL <http://jsantell.github.io/dsp-with-web-audio-presentation/images/am.jpg>. Date Accessed: March 2016.
- [79] Oxford Dictionary. sibilance, January 2016. URL <http://www.oxforddictionaries.com/translate/english-italian/sibilance>. Date Accessed: March 2016.

- [80] acoustics.com. Codes and testing, 2003 - 2009. URL http://www.acoustics.com/codes_testing.asp. Date Accessed: March 2016.
- [81] Rob Shaddick. A developers guide to: Approved Document E Sound Testing, January 2013. URL <http://www.soundguard.co.uk/wp-content/uploads/2013/07/A-Guide-to-Approved-Documents-E-Sound-Testing.pdf>. Date Accessed: March 2016.
- [82] Sulis Acoustics. Sound insulation testing report, April 2011. URL <http://www.jupiterunderfloorheating.com/uploads/knowledgeBase/BJPartETestresultData.pdf>. Project Ref: SA 1165. Date Accessed: March 2016.
- [83] Gustav Theodor Fechner. *Elemente der psychophysik*, 2 bde., 2. unveränd. Aufl., hrsg. von Wilhelm Wundt, Leipzig: Breitkopf und Härtel, 1, 1889.
- [84] Athanasios Papoulis and S Unnikrishna Pillai. *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education, 2002.
- [85] Andrew Abel. Towards an intelligent fuzzy based multimodal two stage speech enhancement system. 2013. PhD Thesis, University of Stirling.
- [86] Reverb challenge, January 2016. URL <http://reverb2014.dereverberation.com/index.html>. Date Accessed: March 2016.
- [87] University of Salford. Comparing methods of testing - objective metrics, January 2015. URL <http://www.salford.ac.uk/computing-science-engineering/research/acoustics/psychoacoustics/sound-quality-making-products-sound-better/sound-quality-testing/sound-quality-testing-objective-metrics>. Date Accessed: March 2016.
- [88] Markus Bodden. Instrumentation for sound quality evaluation. *Acta Acustica united with Acustica*, 83(5):775–783, 1997.
- [89] National Acoustic Laboratories of Australia. Hearing assessment. URL http://www.nal.gov.au/hearing-assessment_tab_objective-testing.shtml. Date Accessed: March 2016.
- [90] Nabila Ahmed. Cochlear heads for earnings record. *The Age. Retrieved*, pages 04 – 27, 2008.

-
- [91] Shih-Chii Liu, Tobi Delbruck, Giacomo Indiveri, Adrian Whatley, and Rodney Douglas. *Event-Based Neuromorphic Systems*. John Wiley & Sons, 2014.
- [92] Paul W Glimcher and Ernst Fehr. *Neuroeconomics: Decision making and the brain*. Academic Press, 2013.
- [93] Scott L Hooper, Kevin H Hobbs, and Jeffrey B Thuma. Invertebrate muscles: thin and thick filament structure; molecular basis of contraction and its regulation, catch and asynchronous muscle. *Progress in neurobiology*, 86(2):72 – 127, 2008.
- [94] Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500 – 544, 1952.